AFIT/GE/ENG/92D-28

COMPUTER SIMULATION OF A
GENERAL PURPOSE SATELLITE MODEM

THESIS

William Lee Montgomery Jr.

Captain, USAF

AFIT/GE/ENG/92D-28

AFIT/GE/ENG/92D-28

COMPUTER SIMULATION OF A GENERAL PURPOSE SATELLITE MODEM

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

William Lee Montgomery Jr., B.S.
Captain, USAF

December 1992

Approved for public release; distribution unlimited

Acknowledgments

The completion of this thesis was a challenge. First, I would like to thank Major Mark Mehalic, my thesis advisor, for his guidance throughout this project. I also appreciate the support given to me by my committee members, Capt Joe Sacchini and Mr Jim Foshee. I would like to thank Mr Brian Trexel for his insights into the workings of the modem and for listening to all my questions about the modem.

On August 27, 1991, my wife Julia had our first child Elizabeth Marie. Elizabeth brought many changes to our family and my study habits. Although Elizabeth is not old enough to know it, she taught me more about life than many years at AFIT ever could. In addition to taking care of Elizabeth, Julia made time to listen to my gripes about my thesis and most importantly, time to type the text in it. Finally, I can not thank Julia enough for her help and love.

Table of Contents

	Page
List of Figures	vii
List of Tables	ix
Abstract	xi
I. Introduction	1
1.1 Background	1
1.1.1 Computer Simulation	1
1.1.2 Monte Carlo Technique	2
1.1.3 CQM-248 Modem Basics	2
1.1.4 Code Gain	2
1.2 Goal	3
1.3 Assumptions	3
1.4 Scope	4
1.5 Approach	5
1.5.1 BPSK Matched Filter System	5
1.5.2 Costas Loop	5
1.5.3 Full System Testing	6
1.6 Overview	6
II. Equipment Description	8
2.1 CQM-248 Digital Satellite Modem	8
2.1.1 Modulator Block Diagram	8
2.1.2 Demodulator Block Diagram	10
2.2 Computer Simulation Tools	14

2.2.1 Block Diagram Editor 14

2.2.2 Signal Display Editor 15

2.3 Simulation at Baseband 15

2.4 Summary 18

III. Theoretical Probability of Bit Error 19

3.1 Probability of Bit Error 19

3.2 Theoretical Code Gain 21

3.3 Methods of Error Estimation 23

3.3.1 Confidence Interval 27

3.3.2 Estimation Error 28

3.3.3 Effect of Dependent Errors 30

3.4 Summary 30

IV. BPSK Matched Filter Test 32

4.1 BPSK Modulation in SPW™ 32

4.2 Data Flow in the System 35

4.3 Timing Considerations 37

4.4 Testing Process 38

4.5 Test Results BPSK Matched Filter System 39

4.6 Summary 40

V. Costas Loop 42

5.1 Costas Loop Demodulation Theory 42

5.2 Summary of Theoretical Operation Costas Loop . . 51

5.3 The SPW™ Costas Loop Block 51

5.3.1 Costas Loop Testing 53

5.3.2 Phase Step Test System and Results . . 53

5.3.3 Frequency Step Test System and Results 55

5.4 Debugging the Costas Loop 56

5.5 Summary of Costas Loop Testing 59

5.5.1 Phase Step 59

5.5.2 Frequency Step 60

5.6 Summary 61

VI. Coded BPSK System Simulation 63

6.1 Encoder and Decoder Block Descriptions 63

6.1.1 Differential Encoder and Decoder . . . 63

6.1.2 Convolutional Encoder Configuration . . 64

6.1.3 Viterbi Decoder Configuration 66

6.1.4 Back-to-Back Testing 68

6.2 Complete System Testing 69

6.2.1 Signal Flow 70

6.2.2 System Parameters 71

6.2.3 Timing Considerations 71

6.3 Testing Process 74

6.3.1 Initial Test Results 76

6.3.2 Five Samples/Encoded Bit Result 77

6.3.3 Additional Testing 78

6.3.3.1 Increased Sample Frequency . . 78

6.3.3.2 Decreased Bit Rate 79

6.3.3.3 Ten Samples/Encoded Bit Test . 80

6.3.4 Ten Samples/Encoded Bit Result 82

6.4 Results Discussion 83

6.5 Summary 84

VII. Conclusions and Recommendations 86

7.1 Conclusions 86

7.2 Recommendations 88

Appendix A 89

Appendix B 115

References 123

Vita 125

List of Figures

Figure		Page
1.	CQM-248 Modulator	9
2.	CQM-248 Demodulator	10
3.	Simplified Costas Loop	11
4.	Demodulator Block Diagram With Costas Loop	12
5.	Theoretical Probability of Bit Error	21
6.	Theoretical Code Gain	23
7.	Example Probability Density Functions [3]	24
8.	Confidence Bands When Observed Value is 10^{-k} [3]	29
9.	BPSK Matched Filter System	33
10.	Simulation Bandwidth of a Gaussian Random Variable	34
11.	Probability of Bit Error	41
12.	Reference Costas Loop	43
13.	Squaring Loop	44
14.	Complete System	69
15.	Theoretical and Measured Error Rate Plots for 5 Samples per Encoded Bit	77
16.	Theoretical and Measured Error Rate Plots for 5 and 10 Samples per Encoded Bit	83
17.	Block Diagram of BPSK Matched Filter System	90
18.	White Noise Generator Detail	91
19.	PSK Matched Filter Demodulator Detail	92
20.	PSK Detector Detail	93
21.	Costas Loop Detail	94
22.	Modified Costas Loop Detail	95
23.	Costas Loop Symbol	96

24. Costas Loop Phase Step Test System 97

25. Costas Loop Frequency Step Test System 98

26. Costas Loop Response to Phase Step Plot 1. 99

27. Costas Loop Response to Phase Step Plot 2. 100

28. Costas Loop Response to Phase Step Plot 3. 101

29. Costas Loop Response to Phase Step Plot 4 102

30. Costas Loop Response to Frequency Step Plot 1 103

31. Costas Loop Response to Frequency Step Plot 2 104

32. Costas Loop Response to Frequency Step Plot 3 105

33. Costas Loop Response to Frequency Step Plot 4 106

34. Differential Encoder Detail 107

35. Differential Encoder Symbol 108

36. Differential Decoder Detail 109

37. Differential Decoder Symbol 110

38. Full System for Timing Verification 111

39. Full System Used for Five Samples per Coded Bit Test . . 112

40. Test Ten Samples per Coded Bit at 2400 Bit per Second . 113

41. Test Ten Samples per Coded Bit at 1200 Bit per Second . 114

List of Tables

Table		Page
1.	Predicted Code Gain	22
2.	Test Parameters	38
3.	Theoretical P_b	39
4.	Noise Seeds	39
5.	$\hat{P}_b$ Statistics Matched Filter Detection BPSK	40
6.	Theoretical Costas Loop Parameter Values	52
7.	Phase or Frequency Step Test Parameters	54
8.	Phase Step Test Signal Sinks	55
9.	Frequency Step Test Signal Sinks	57
10.	Costas Loop Test Result For VCO Gain 65.087382 Hz/V and a Phase Step Input	59
11.	Costas Loop Test Result For VCO Gain 130.17476 Hz/V and a Phase Step Input	60
12.	Costas Loop Frequency Step Test Result	60
13.	System Parameters For 5 Samples/Encoded Bit Test	72
14.	Number of Iterations For 5 Samples/Encoded Bit Test	75
15.	$\hat{P}_b$ Statistics For 5 Samples/Encoded Bit	76
16.	Code Gain For 5 Samples/Encoded Bit Test	78
17.	$\hat{P}_b$ Statistics For 48000 Hz Sample Frequency: 3.5 dB E_b/N_o	79
18.	$\hat{P}_b$ Statistics For 1200 bps Data Rate: 3.5 dB E_b/N_o	80
19.	Number of Iterations For 10 Samples/Encoded Bit Test	81
20.	System Parameters For 10 Samples/Encoded Bit Test	81
21.	$\hat{P}_b$ Statistics For 10 Samples/Encoded Bit	82
22.	Code Gain For 10 Samples/Encoded Bit Test	83

23. $\hat{P}_b$ Matched Filter Detection BPSK For 4 dB E_b/N_o 115

24. $\hat{P}_b$ Matched Filter Detection BPSK For 6 dB E_b/N_o 115

25. $\hat{P}_b$ Matched Filter Detection BPSK For 8 dB E_b/N_o 116

26. $\hat{P}_b$ Statistics Matched Filter Detection BPSK 116

27. $\hat{P}_b$ For 5 Samples/Encoded Bit: 3.5 dB E_b/N_o 117

28. $\hat{P}_b$ For 5 Samples/Encoded Bit: 4.5 dB E_b/N_o 117

29. $\hat{P}_b$ For 5 Samples/Encoded Bit: 5.0 dB E_b/N_o 118

30. $\hat{P}_b$ For 5 Samples/Encoded Bit: 5.5 dB E_b/N_o 118

31. $\hat{P}_b$ Statistics For 5 Samples/Encoded Bit 119

32. $\hat{P}_b$ For 48000 Hz Sample Frequency: 3.5 dB E_b/N_o 119

33. $\hat{P}_b$ For 1200 bps Data Rate: 3.5 dB E_b/N_o 120

34. $\hat{P}_b$ Statistics For 1200 bps Data Rate: 3.5 dB E_b/N_o 120

35. $\hat{P}_b$ For 10 Samples/Encoded Bit: 4.5 dB E_b/N_o 121

36. $\hat{P}_b$ For 10 Samples/Encoded Bit: 5.0 dB E_b/N_o 121

37. $\hat{P}_b$ For 10 Samples/Encoded Bit: 5.5 dB E_b/N_o 122

38. $\hat{P}_b$ Statistics For 10 Samples/Encoded Bit 122

AFIT/GE/ENG/92D-28

Abstract

The purpose of this research was to model and simulate the performance of a digital phase shift keyed satellite modem. The probability of bit error (P_b) at different levels of energy per bit to noise power ratio (E_b/N_o) was the performance measure. The channel was assumed to contribute only additive white Gaussian noise.

A second order Costas loop performs demodulation in the modem and was the key part of the simulation. The Costas loop with second order Butterworth arm filters was tested by finding the response to a phase or frequency step. The Costas loop response was found to be in agreement with theoretical predictions in the absence of noise. Finally, the effect on P_b of a rate 1/2 constraint length 7 convolutional code with eight level soft Viterbi decoding was demonstrated by the simulation. The simulation results were within 0.7 dB of theoretical.

All computer simulations were done at baseband to reduce simulation times. The Monte Carlo error counting technique was used to estimate P_b. The effect of increasing the samples per bit in the simulation was demonstrated by the 0.4 dB improvement in P_b caused by doubling the number of samples.

COMPUTER SIMULATION OF A GENERAL PURPOSE SATELLITE MODEM

I. Introduction

1.1 Background

The modern world of communications is a rapidly changing environment. In the area of satellite communications, systems using digital signal processing (DSP) are increasingly found. The CQM-248 modem is a modem that uses DSP to implement many functions [1]. As communications system complexity has grown, the analytical evaluation of the system performance has become difficult if not impossible [2]. To attempt to predict system performance, engineers turn to computer simulations of a system model. The criteria often used for system performance measurement is the probability of bit error (P_b).

1.1.1 Computer Simulation.

The availability of simulation packages has made the use of computers to simulate and evaluate the performance of communication systems commonplace [3]. The use of an interactive package relieves the engineer from the tedious job of computer programming and calculation of complex, often repetitive, calculations.

The computer simulation is a mathematical model of the communications system. The computer performs calculations on the independent inputs (data bits) and produces output (data bits) based on the model. Each continuous time data bit is represented in a sampled form in the simulation. Direct comparison of system input and output data bits for differences can be used to estimate P_b.

1.1.2 Monte Carlo Technique. The Monte Carlo technique is an error counting technique used to estimate P_b [3]. To obtain an error rate estimate that approaches the true P_b, many data bits must be compared. The rule of thumb is that a minimum of 10^{k+1} bits must be observed to estimate a true P_b on the order of 10^{-k} [2,3]. The 10^{k+1} bits define a confidence interval where the estimated P_b will range from one half to twice the true P_b. This interval is considered acceptable [3].

1.1.3 CQM-248 Modem Basics. The modem is designed to implement phase-shift keyed modulation (PSK) on digital data [1]. The digital data is scrambled, differential, and convolutional encoded before wave shaping occurs. The wave shaping is done by proprietary application-specific integrated circuits and is not considered in this thesis. The modem has both sequential and Viterbi decoding capabilities for demodulation of the encoded data. The modem is capable of producing either binary PSK (BPSK) or quadrature PSK (QPSK) forms. Options are also available to implement 8-PSK with trellis coded modulation and off-set QPSK (OQPSK).

1.1.4 Code Gain. The use of coding schemes in communication systems has become an accepted practice [4]. Convolutional coding with soft decision Viterbi decoding is a standard technique used over satellite communication channels.

Performance of Viterbi decoding with eight level quantized soft decisions has been shown to be within 0.25 dB of theoretical predictions [5]. For a rate 1/2 constraint length 7 convolutional code with ideal BPSK detection, the code gain predicted by [4] is 3.8 dB for $P_b = 1X10^{-3}$.

1.2 Goal

The goal of this thesis is to estimate the CQM-248 modem performance. This will be accomplished through the use of computer modeling and simulation. The probability of bit error in the presence of additive white Gaussian noise is the performance measure. The Monte Carlo technique is used to estimate the probability of bit error. Specifically, the Signal Processing Worksystem™ (SPW™) software tool is used to model the communications system. To accomplish this goal, three separate objectives will be pursued. The first objective is to gain experience with the SPW™ tool. The second objective is to develop and demonstrate proper operation of a Costas loop demodulator. The final objective is to assemble and test a full system model.

1.3 Assumptions

Five assumptions affecting the modem simulation are made.

1. The effects of wave shaping on P_b will be neglected. Due to the proprietary wave shaping used in the CQM-248, no attempt is made to model wave shaping.

2. Perfect symbol synchronization is achieved.

The symbol synchronization used by the modem is a slope detection technique that works on shaped pulses [6], and because shaped pulses are not modeled, any attempt at modeling symbol synchronization is unjustified. Symbol synchronization is accomplished in the simulation by use of simulation generated timing signals.

3. The simulation random data generator is assumed to be sufficiently random to eliminate the scrambler part of the modem from the model.

4. The automatic gain control (AGC) portion of the modem is assumed to negligibly affect probability of bit error and is not included in the model. It has been shown that variation of AGC gain of ±20% is tolerated by Viterbi decoding (soft decision) with no significant performance degradation [5].

5. Finally, no attempt will be made to model the modem signal acquisition processes.

1.4 Scope

The simulation system is used to model two modems operating over a channel that disturbs the transmitted signal with additive white Gaussian noise (AWGN). All simulation is done at baseband due to the long simulation run time involved at actual operating frequencies. Bandpass systems can be successfully modeled with their lowpass equivalent model, thus reducing the simulation sample frequency and computer run time [2].

The modem configuration modeled is the differentially encoded BPSK (DEBPSK) option. Data will have both differential and convolutional encoding applied. The convolutional code used is the optimum rate 1/2 constraint length 7 code [4]. The demodulation process of the modem is modeled as a second-order Costas loop. The probability of bit error will be estimated for four energy per bit to noise power (E_b/N_o) levels. The lowest E_b/N_o level tested is 3.5 dB, based on practical experience with the modem [6]. Also, the probability of bit error at lower levels of E_b/N_o for the coded system rapidly approaches uncoded BPSK performance [4].

1.5 Approach

1.5.1 BPSK Matched Filter System. The first objective is to gain experience with the SPW™ tool. To do this, a basic BPSK matched filter detection system will be modeled using standard SPW™ system blocks. The AWGN added by the channel will be used to generate bit errors. The Monte Carlo method of estimating P_b will be demonstrated. The average of ten simulations, each run with a different noise seed, will be found for three E_b/N_o levels. The resulting data will be plotted and compared to theoretical P_b for matched filter detection of BPSK. There are three goals to this portion of testing. First, the Monte Carlo method is demonstrated. Second, operation of the bit error rate counter is verified. Finally, the AWGN generator block is confirmed by its ability to cause errors in the system at the expected rate.

1.5.2 Costas Loop. The second objective is to develop and demonstrate the proper performance of a Costas loop demodulator. The Costas loop is the key part of the demodulation process. Two test systems will be built to demonstrate correct Costas loop operation. First, a system will be built to test the loop response to a phase step input. The loop phase error rise time to peak phase deviation and magnitude of the deviation are the performance criteria. Theoretical values for rise time and peak deviation of the phase error are predicted by [7]. The second test of the Costas loop will be its response to a frequency step input. This is commonly used to represent a Doppler shift in the carrier frequency [7]. The time to lock onto the frequency step and steady-state phase error are the criteria used for the test. Again, [7] provides the theoretical values.

1.5.3 Full System Testing. The final objective is to assemble and test a full system. The convolutional encoder block configured to model the modem will encode the differentially encoded random data. The encoded data will be BPSK modulated at baseband and AWGN added to simulate the channel. The Costas loop will demodulate the noisy data which will then be soft Viterbi decoded. The error counter will compare the decoded data to the original random data and total the errors. A curve of P_b vs E_b/N_o will be produced from the measured data and plotted for comparison to theory.

1.6 Overview

The thesis is organized in seven chapters. Chapter II has two main parts. First, a hardware description of the CQM-248 modem is provided. Then, an overview of the software tool used to build the modem model is included and simulation at baseband is discussed. Chapter III covers theoretical probability of bit error for BPSK and DEBPSK. Also, theoretical code gain is given. The chapter ends with a section on Monte Carlo error estimation. Chapter IV demonstrates the Monte Carlo method applied to matched filter detection of BPSK. Two of the goals in Chapter IV are to test the bit error counter and AWGN generator blocks. Chapter V provides Costas loop theory and results of testing the simulation Costas loop. Chapter VI describes the modem model testing and results. Chapter VII is a summary of the thesis work and conclusions. Appendix A includes SPW™-generated system plots and signal plots. Finally, Appendix B contains data tables from the computer simulation runs.

II. Equipment Description

2.1 CQM-248 Digital Satellite Modem

The CQM-248 is a digital PSK modem used in satellite communication. The modem serves as a link between the user's baseband data terminal equipment and the intermediate frequency (IF) of the radio equipment. It is constructed with a modular architecture to allow selection of operating configuration and ease of maintenance [1]. A simplified modulator block diagram is shown in Figure 1, with the demodulator block diagram shown in Figure 2. The next two sections discuss the blocks included in the simulation.

2.1.1 Modulator Block Diagram. The modulator consists of nine blocks as follow: data scrambler, differential encoder, convolutional encoder, digital filter, digital-to-analog converter, modulator, IF synthesizer, power control, and bandpass filter. The data is first scrambled to reduce the chance of transmitting a string of zeros or ones. For the simulation, a random data generator is used as the input, so the scrambler block is not included. The modem uses differential encoding to eliminate the 180 degree phase ambiguity. Differential encoding is included in the simulation. The next block in the data path is the convolutional encoder. The CQM-248 can implement two code rates with a constraint length of either 7 or 9. The code rates are 1/2 or 3/4. Only the rate 1/2 constraint length 7 is simulated. The generator functions of the convolutional code are denoted as G_0 = 1111001 (binary) and G_1 = 1011011 (binary) [6].

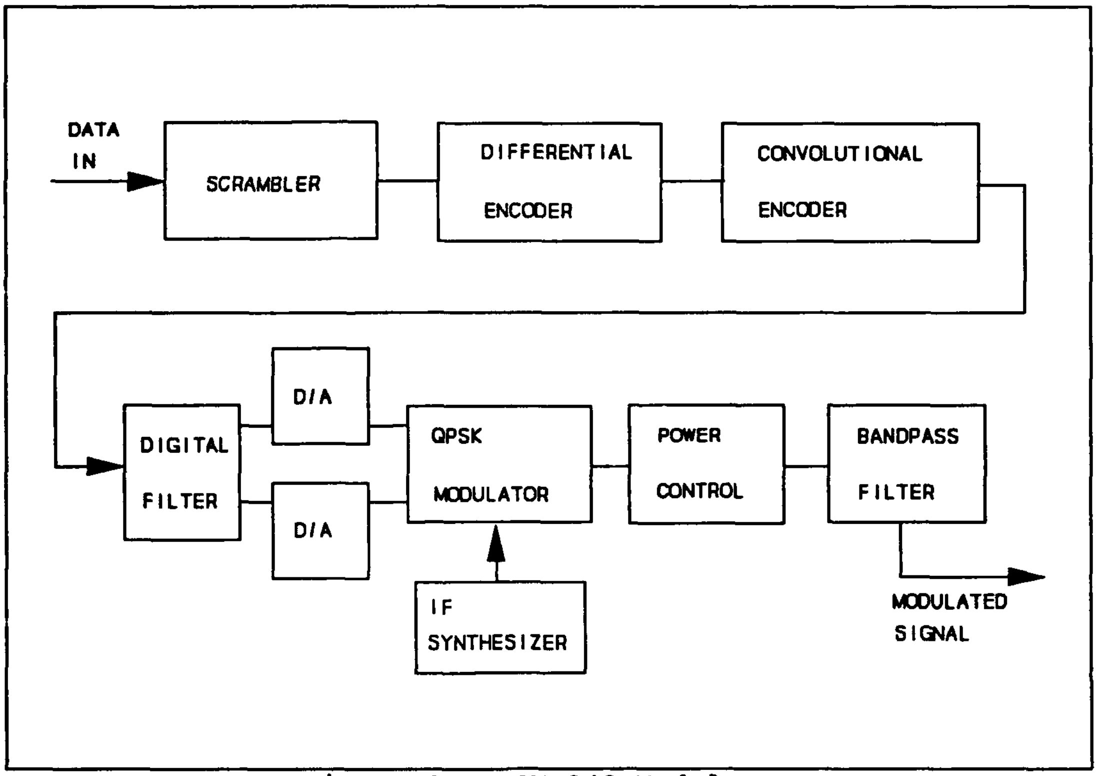

Figure 1. CQM-248 Modulator

These functions are commonly known as the optimum functions for a rate 1/2 constraint length 7 code [8]. A SPW™ convolution encoder block was configured to model this portion of the modulator.

The next block in the data path is the wave shaping digital filter. Due to the proprietary nature of the wave shaping, the information needed to simulate this block was not available. Therefore, it is not simulated. The output from the filter stage is an in-phase and quadrature signal that is converted from digital to analog and sent to the QPSK modulator block. The digitally synthesized carrier is modulated by the antipodal data stream. To transmit a BPSK signal, the same data bits are sent to both the in-phase and quadrature channel of the modulator. For the simulation, a 0 Hz carrier is used and only BPSK modulation is simulated.

The PSK modulated signal power is set by a power control block. Finally, a bandpass filter bandlimits the signal. Neither the power control or bandpass filter are implemented in the modem model.

2.1.2 Demodulation Block Diagram. Figure 2 is a simplified demodulator block diagram. For reference, a simplified block

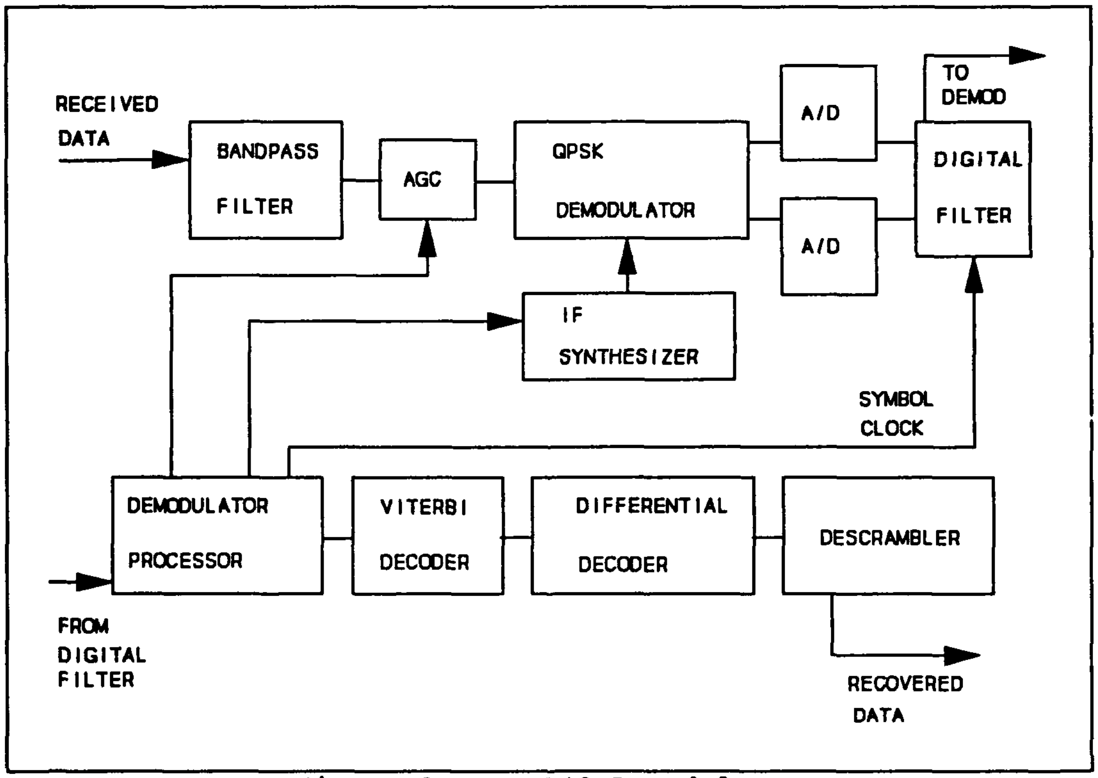

Figure 2. CQM-248 Demodulator

diagram for a Costas loop is shown in Figure 3. Six modem blocks are incorporated into the Costas loop. The blocks included are

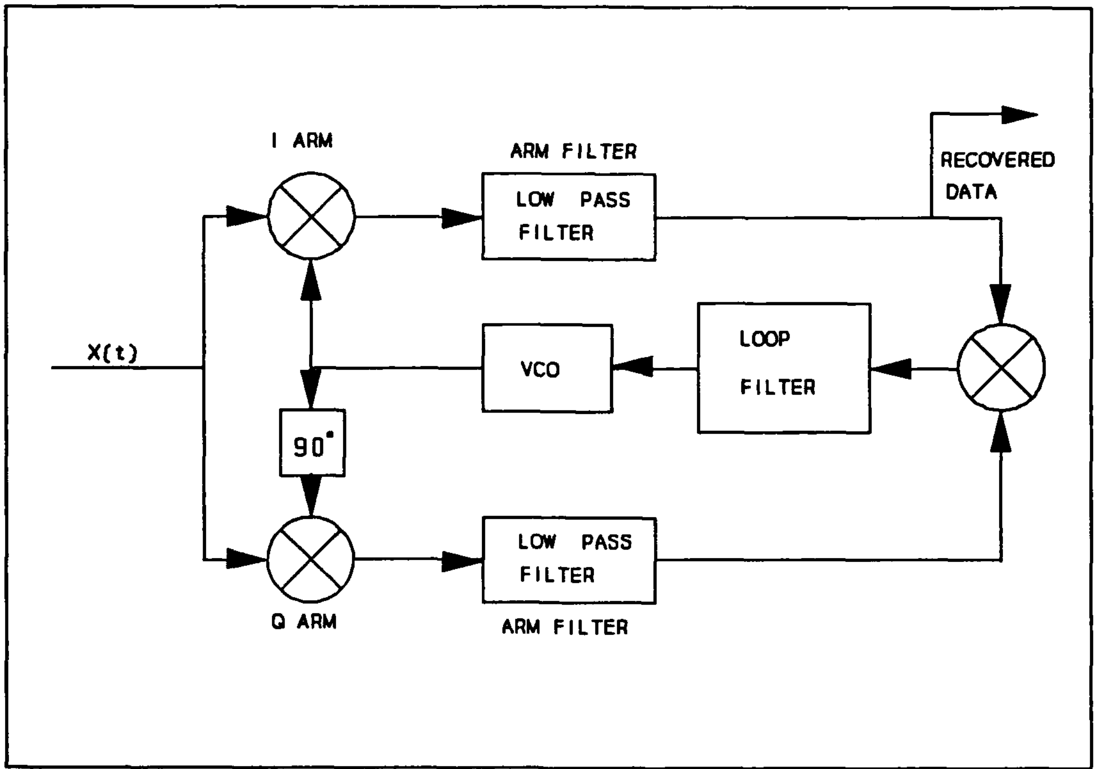

Figure 3. Simplified Costas Loop

the QPSK demodulator; IF synthesizer; analog to digital converters; digital filter; and demodulator processor. Figure 4 shows where the Costas loop fits into the modem data flow. The input filter is a bandpass filter capable of passing the range of intermediate frequencies available from the CQM-248. The range of frequencies is 52 MHz to 88 MHz or 104 MHz to 176 MHz [1]. This filter is not included in the model because the bandwidth of the digital filter accomplishes the noise spectrum limiting [6]. Also, the automatic gain control (AGC) is not included in the model because it is shown by [4] to have little effect on bit error rate for a ±20% range of AGC.

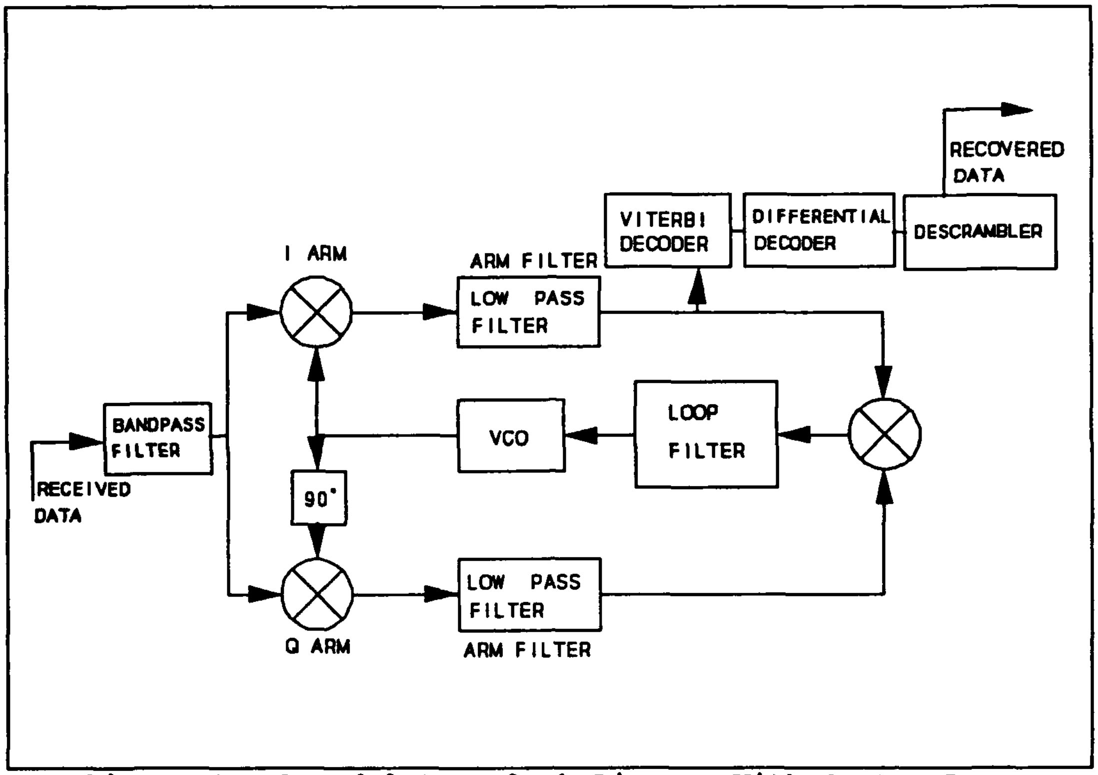

Figure 4. Demodulator Block Diagram With Costas Loop

The modem block QPSK demodulation corresponds to the in-phase (I) arm and quadrature (Q) arm mixers of the Costas loop. The IF synthesizer can be considered the voltage controlled oscillator (VCO) of the Costas loop. The analog-to-digital conversion of the modem is not simulated because the simulation is all based on sampled data bits.

The digital filter in the modem is a patented application-specific integrated circuit. The digital filter is programmable to realize a variety of equalized filter shapes for data rates from below 9.6 Kbps to above 2.2 Mbps [1]. The demodulator processor is another application-specific integrated circuit. The demodulator process performs both carrier phase detection and symbol synchronization, providing control signals to the digital synthesizer and closing the loop of the carrier tracking circuit. Two other control outputs come from the demodulator processor: a digital control signal used by the AGC circuits, and a symbol synchronization sent to the digital filter and provided as a modem output.

For the model, 2-pole Butterworth lowpass filters were selected for the Costas loop arm filters for three reasons. The Butterworth arm filter performance in a Costas loop is well documented in [9]. The modem digital filter information is unavailable. Lastly, the SPW™ Costas loop block was available with Butterworth arm filters. Phase detection is accomplished digitally in the modem and will be modeled as a mixer. The loop filter in the demodulator processor is a proportional plus integrator circuit that can be approximated by a lead-lag filter [7]. The digital synthesizer is modeled as a VCO. A full discussion of the Costas loop demodulator used in the simulation is provided in Chapter V.

The data output by the demodulator processor is a quantized representation of the analog input. Both the modem digital filter and demodulator processor affect the quantization level. The final result can be modeled as a eight level quantized ±1 signal with (-1, -0.66, -0.33, 0, 0.33, 0.66, 1) as the thresholds [6]. The quantized data bits are used to perform soft Viterbi decoding, and the Viterbi decoded data is the input to the differential decoder block. A SPW™ Viterbi decoder block was configured to model the modem Viterbi decoder.

A differential decoder was included in the model. As previously stated, a random data generator was used as an input to the system; therefore, both data scrambler and descrambler were not included in the model.

2.2 Computer Simulation Tools

The Signal Processing WorkSystem™ and SPW™ are trademarks of Comdisco Systems, Inc. [10]. The SPW™ software package provides an interactive computer-aided tool for digital signal processing simulation. The two major parts of SPW™ that were used in this thesis are the Block Diagram Editor (BDE) and the Signal Display Editor (SDE). Two copies of the software tool were available for use. One copy ran on a Sun-4™ and the other copy ran on a SPARC-2™. The complied C code from the simulation was also run on three SPARC-2™ workstations.

2.2.1 Block Diagram Editor. The BDE is used to create and edit systems consisting of signal processing blocks. Signal processing blocks are stored in libraries. The graphical interface of the BDE allows graphic representations of the blocks called symbols to be connected to form a system. Multiple symbols can be joined together as a detail and a new symbol created that performs the function of the detail. Signal processing by a symbol can be controlled by editing the symbol parameters. Parameters control such operating functions as sample frequency or bit rate. Parameters can be set to a desired value or calculated from other parameters. For example, a gain value in a feedback loop can be set to -10 or calculated to be -10 from the equation: -(sample frequency/bit rate).

A completed system with parameters set to the desired values or equations can be interactively simulated from the BDE. When the simulation run is started, the number of samples to process (iterations) is entered and the default noise seed of 1 is used. The results of the simulation run are stored in files in two ways. The signal sink block in a system will store numeric values in a file that can be displayed by the SDE. The write results block stores the block instance number and iteration number along with the system result to a file. The write results block output can be viewed in a text window in the BDE or with a text editor on the computer.

2.2.2 Signal Display Editor. The SDE software module is used to create, edit, display, and analyze signal waveforms. Signals stored from a simulation run can be overlaid to check timing. Also, math functions can be performed on the stored signals to normalize results. The SDE allows all iterations to be displayed at one time or a few iterations to be zoomed in on to find a peak value. Typical plots from the SDE are shown in Appendix A.

2.3 Simulation at Baseband

In general, a modulated carrier can be represented in quadrature form given by [11] as:

$$S(t) = X_1(t)\cos(2\pi f_c t + \theta) - X_2(t)\sin(2\pi f_c t + \theta) \qquad (1)$$

where

$X_1(t)$ and $X_2(t)$ are low-pass processes

f_c is the carrier frequency

B is the bandwidth of the low-pass process

f_c is typically much greater than B, and

θ is the carrier phase offset also called channel rotation

The complex envelope form of Equation 1 is

$$S(t) = [X_1(t) + jX_2(t)]e^{j\theta} \qquad (2)$$

where $f_c = 0$ Hz.

In the simulation, the continuous time signal $S(t)$ is processed as a sampled signal $S(k\Delta t)$ with a sampling frequency $f_{sa} \doteq 1/\Delta t$, and k is the particular sample number. In SPW™, the sample number is referred to as the iteration number. The representation of $S(t)$ for a given sampling frequency f_{sa} is given as [2]:

$$S(k\Delta t) = A(k\Delta t)\sqrt{P}e^{\left\{j2\pi\left(\frac{f_c}{f_{sa}}\right)k + j\Phi(k\Delta t)\right\}} \qquad (3)$$

where

$A(k\Delta t)$ represents amplitude modulation

$\Phi(k\Delta t)$ represents phase modulation

$\left(\frac{f_c}{f_{sa}}\right)$ is the relative carrier frequency

P is the signal power

$f_{sa} = \frac{1}{\Delta t}$ is the sampling frequency, and

k is the iteration index

For BPSK, $S(k\Delta t)$ will have the form [2]:

$$S(k\Delta t) = A(k\Delta t)\sqrt{\frac{P}{2}}e^{\left\{j2\pi k\left(\frac{f_c}{f_{sa}}\right) + j\theta\right\}} \tag{4}$$

where

A(kΔt) is a random ±1 modulating bit stream, and

θ is a constant channel rotation

For the baseband model, f_c = 0 Hz. The sampling rate can now be set based on the modulating symbol rate and the effects of aliasing on the power spectra density. As discussed in [2], the f_{sa} producing the best results will be an even integer 8 to 16 times the symbol rate. If less than 8 samples per symbol are used, accuracy is lost, and with more than 16 samples per symbol, simulation time may become excessive with little gain in accuracy [2]. For simulation of the modem, f_{sa} is set to 10 times the bit rate.

2.4 Summary

This chapter provided an introduction to the modem to be modeled. The data flow through the modem was discussed, and the major SPW™ blocks used in the simulation related to modem blocks. A major point was the way the Costas loop incorporates blocks from the modem. Also, the fact that information needed to simulate the proprietary wave shaping was not available was mentioned. The Block Diagram Editor and Signal Display Editor from the software tool used for the simulation were presented. Finally, computer simulation at baseband considerations are presented, and the method used to select the sampling frequency was discussed. A sampling frequency ten times the bit rate will be used in simulation.

III. Theoretical Probability of Bit Error

This chapter presents the theoretical probability of bit error, P_b. Theoretical curves for P_b vs energy per bit to noise power (E_b/N_o) ratio are plotted for binary phase shift keying (BPSK) and differentially encoded BPSK (DEBPSK). Next, the theoretical code gain for convolutional encoding is reviewed. A plot of P_b vs E_b/N_o resulting from the convolutional encoding is provided. Then, the effect of imperfect carrier reference on the P_b curve is discussed. The chapter ends with the theory behind the Monte Carlo method of P_b estimation. The confidence interval associated with the use of the Monte Carlo P_b estimate is presented.

3.1 Probability of Bit Error

A common measure of performance used for comparing digital modulation methods is the probability of symbol error (P_e) [8]. For BPSK, the probability of symbol error is the same as the probability of bit error P_b. For coherent detection of BPSK with antipodal signaling, the equation for P_b is given in [8] as

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_o}}\right) \tag{5}$$

where

E_b is the energy per received bit

N_o is the white noise power

$Q(\cdot)$ is the complementary error functions defined by

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} e^{\left(\frac{-t^2}{2}\right)} dt \qquad (6)$$

The energy per bit is [8]

$$E_b = \int_0^T S_1^2(t)\, dt = \int_0^T S_2^2(t)\, dt \qquad (7)$$

For antipodal BPSK, where S(t) is the bit amplitude ±A, and $S_1^2(t) = S_2^2(t) = S^2(t)$, E_b reduces to

$$E_b = \int_0^T A^2 dt = A^2 T \qquad (8)$$

When suppressed carrier recovery methods are used for BPSK carrier recovery, a ±180° phase ambiguity exists [8]. Differential encoding is a common method used to avoid the phase ambiguity. The probability of bit error for coherently detected, DEBPSK is given by [8].

$$P_b = 2Q\left(\sqrt{\frac{2E_b}{N_o}}\right)\left[1 - Q\left(\sqrt{\frac{2E_b}{N_o}}\right)\right] \qquad (9)$$

Figure 5 is a plot of Equation 5 and Equation 9. The plot was computer generated for 1/4 dB steps of the independent variable E_b/N_o.

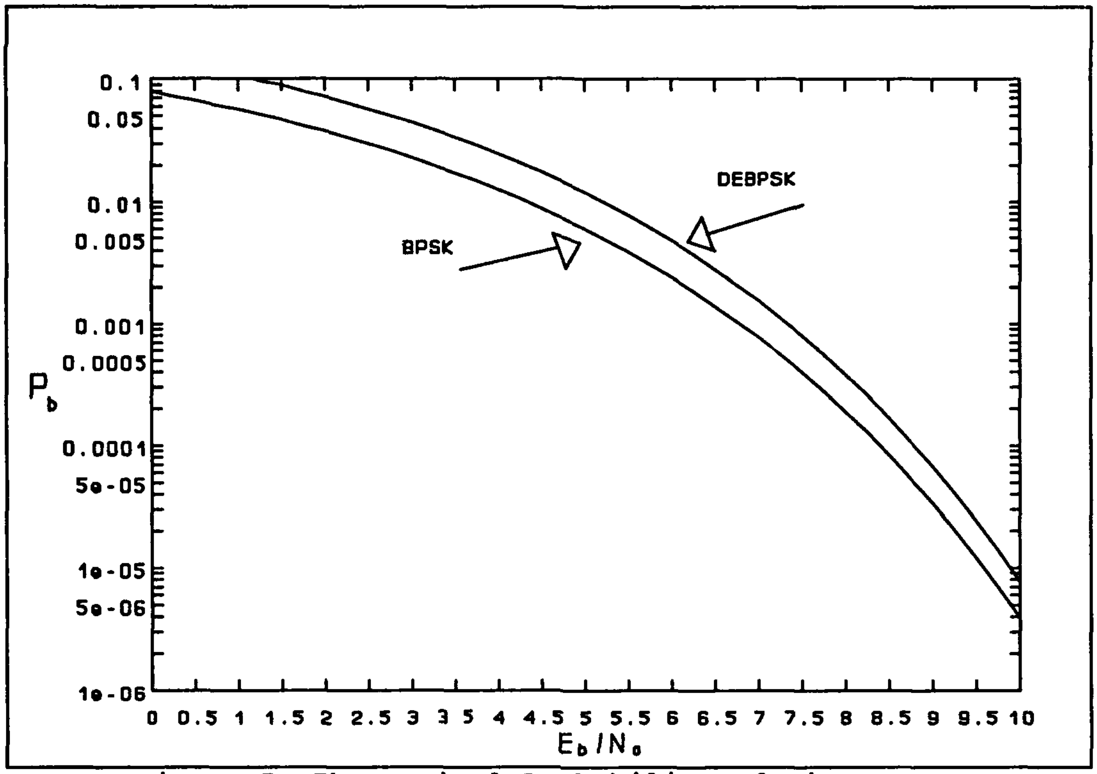

Figure 5. Theoretical Probability of Bit Error

3.2 Theoretical Code Gain

The effect of using a convolutional code on the bit stream before BPSK modulation and soft decision Viterbi decoding on the demodulated bit stream is discussed in [4,5]. The desired result of coding is to reduce the amount of signal energy required to receive data at a specified P_b. The amount of reduction of E_b/N_o required to achieve a given P_b is defined as the code gain [4]. For the rate 1/2, constraint length 7 code with the same generator polynomials used in the modem, [4,5] predict a coding gain shown in Table 1.

Table 1. Predicted Code Gain

P_b	Code Gain
10^{-3}	3.8 dB
10^{-5}	5.5 dB
10^{-7}	5.8 dB
Upper Bound	7.0 dB

This predicted gain is from simulation with equally spaced eight level quantization for (-1.5, -1, -0.5, 0, 0.5, 1.0, 1.5) thresholds. The path length of the Viterbi decoder was 32. According to [4,5] the eight level quantization suffers a loss of less than 0.25 dB compared to infinitely fine quantization. Also, a path length in the decoder of 4 to 5 times the code constraint length is sufficient for negligible variance from optimum decoder performance.

In Figure 6, the curve labeled DEBPSK with coding was made by subtracting the predicted code gain from the theoretical P_b curve generated from Equation 9. The theoretical code gain curve will be compared to the simulation performance.

The theoretical code gain assumes perfect carrier phase synchronization. In real systems an imperfect carrier phase reference degrades the P_b performance of the system. A coded system is particularly sensitive to tracking errors due to the steepness of the P_b versus E_b/N_o curves [5]. For suppressed carrier recovery by a second-order phase-locked loop, the loop signal-to-noise ratio (ρ_1) effect on P_b was shown in [5] to degrade the performance by approximately 0.5 dB for $P_b = 10^{-5}$ and $\rho_1 = 13.5$ dB. The degradation is measured from the $\rho_1 = \infty$ result.

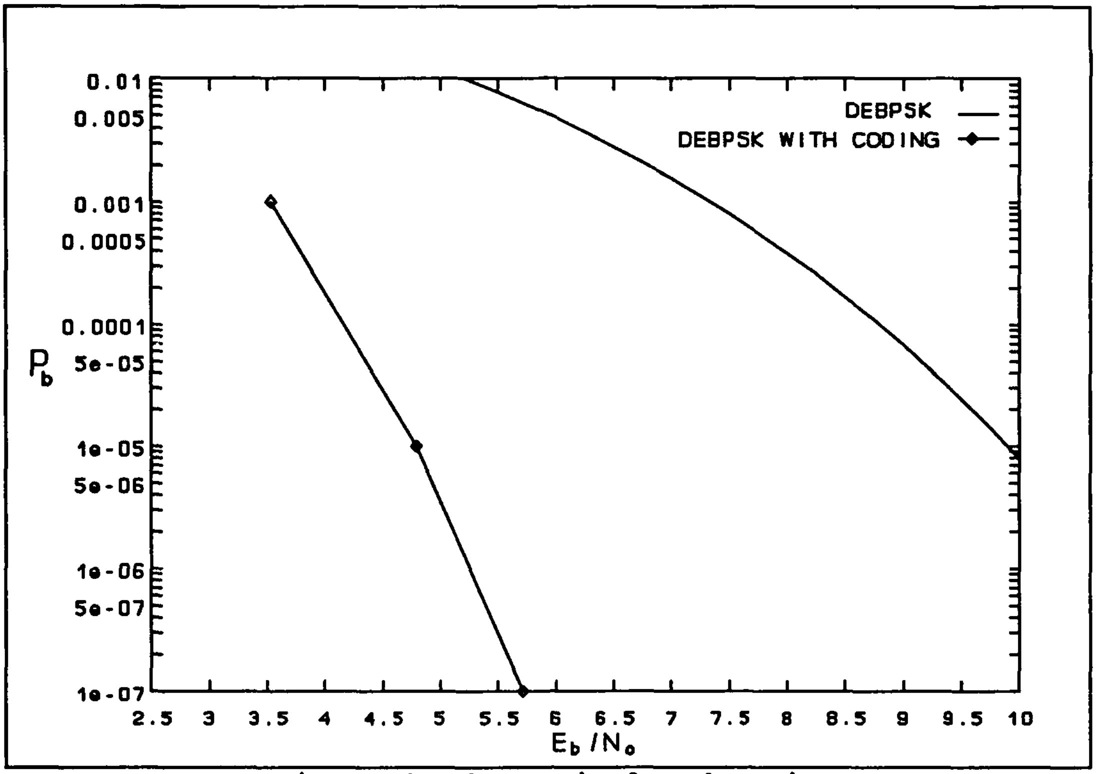

Figure 6. Theoretical Code Gain

For ρ_1 = 12 dB at $P_b = 10^{-5}$, the degradation was shown to be approximately 3 dB, whereas for ρ_1 = 15 dB, the degradation was less than 0.5 dB.

3.3 Methods of Error Estimation

A method of error estimation is required for many digital systems because of nonlinear operations in the demodulation process [2]. For example, the quantizing of the sampled signals and squaring of the signal for carrier recovery are both nonlinear process.

The demodulation and detection process reduces the waveform to a number which is compared to a threshold. The decision process can be described in terms of the probability density functions (pdf) $f_o(V;\tau)$ and $f_1(V;\tau)$ of the input voltage at the sampling instant τ. The decision that a "one" or a "zero" was sent is made based on the input voltage at the sampling time τ. Two possible errors exist. The decision that a "one" was received when a "zero" was sent indicating a large positive excursion of the received voltage from a value of a "zero". Similarly, a sufficient error in the value of a "one" will cause the error in decision resulting in the received value being declared a "zero". Sample pdf plots are shown in Figure 7.

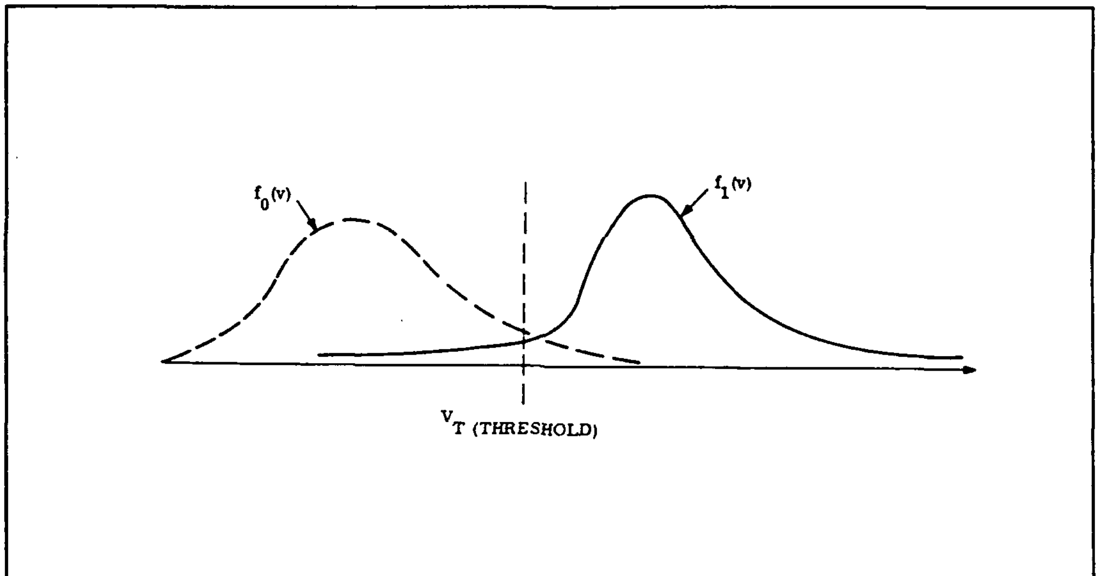

Figure 7. Example Probability Density Functions [3]

In the most general sense, $f_o(V)$ does not have to equal the shape of $f_1(V)$. To simplify notation, the τ dependence is dropped. The probability of error, given that a "one" was sent, is

$$\text{Prob[error/one]} \doteq p_1 = \int_{-\infty}^{V_T} f_1(v)\,dv \tag{10}$$

The probability of error, given that a zero was sent, is

$$\text{Prob[error/zero]} \doteq p_o = \int_{V_T}^{\infty} f_0(v)\,dv \tag{11}$$

The average probability is then

$$p = \pi_1 p_1 + \pi_o p_o \tag{12}$$

where

π_1 is the *a priori* probability of the symbol "one"

π_o is the *a priori* probability of the symbol "zero"

Five methods of making an estimate of the bit error rate are:

a) Monte Carlo simulation

b) modified Monte Carlo simulation

c) extreme-value theory

d) tail extrapolation

e) quasi-analytical (simulation run without noise combined with analytical representation of noise).

The methods vary in their assumptions about and methods for working with the pdfs. Because the Monte Carlo method makes no *a priori* assumptions about $f_o(V)$ and $f_1(V)$ and is the most general of the five methods, it was selected for use in the simulations. Also, the nonlinearity introduced by the quantization in the soft Viterbi decoder makes analytical methods impractical. For a in-depth discussion of the methods, see references [3,11].

The Monte Carlo method relies on error counting to estimate the probability of bit error. Assume a "zero" was sent. The probability of error is

$$p_o = \int_{V_t}^{\infty} f_o(v)\,dv \tag{13}$$

Now consider an error detector defined by

$$h_o(v) = \begin{cases} 1, & v \geq V_T \\ 0, & v < V_T \end{cases} \tag{14}$$

This allows Equation 13 to be rewritten as

$$p_o = \int_{-\infty}^{\infty} h_o(v)\, f_o(v)\,dv \tag{15}$$

Equation 15 can be viewed as the expected value of h_o.

$$p_o = E[h_o(v)] \tag{16}$$

The probability of error can be estimated with a sample mean of h_o.

$$\hat{p}_o = \frac{1}{N}\sum_{i=1}^{N} h_o(v_i) \tag{17}$$

where

h_o is the error detector

The summation is an error counter

$V_i=V(t_i)$, t_i; is the instant of decision time

i is the i th symbol.

In simple terms, N symbols are checked for error and the total error η is add up. Equation 17 defines the Monte Carlo method [3]. Equation 17 can be extended to account for errors when a "one" was sent, but detected incorrectly. Letting η represent the total number of errors out of N bits observed defines an unbiased Monte Carlo estimate of probability of bit error $\hat{P}_e$

$$\hat{p}_e = \frac{\eta}{N} \tag{13}$$

The probability estimate of bit error $\hat{P}_e$ approaches the true error probability p as the number of bits observed N approaches ∞.

<u>3.3.1 Confidence Interval.</u> The Monte Carlo method allows estimating the true probability of error p within a specified confidence interval. After the confidence interval is selected,

the number of bits that must be processed through the system simulation can be found for a desired true value P_e. The confidence level is defined as [3]

$$p[h_2 \leq p \leq h_1] = (1-\alpha) \tag{19}$$

where

$h_1 - h_2$ is the confidence interval

$(1-\alpha)$ is the confidence level

The true value of the bit error rate p will lie between h_1 and h_2 with probability $(1-\alpha)$. Figure 8 shows the confidence interval for three confidence levels (90%, 95%, 99%). Recalling the rule of thumb from Chapter I that N should be on the order of $10/P_e$, the horizontal axis of Figure 8 would be entered at the $N = 10^{k+1}$ point. The 90% confidence level at the $N = 10^{k+1}$ point produces a confidence interval of approximately $0.5p$ to $2p$. This interval is considered acceptable [3].

3.3.2 Estimation Error. The normalized error of the estimated $\hat{P}_e$ is defined as [12]

$$\epsilon = \frac{\text{standard deviation of } \hat{P}_e}{P_e} \tag{20}$$

The true bit error probability in the system is P_e and can be calculated analytically for some cases. An estimator is considered acceptable if the normalized error ε is less than 1.0 [12].

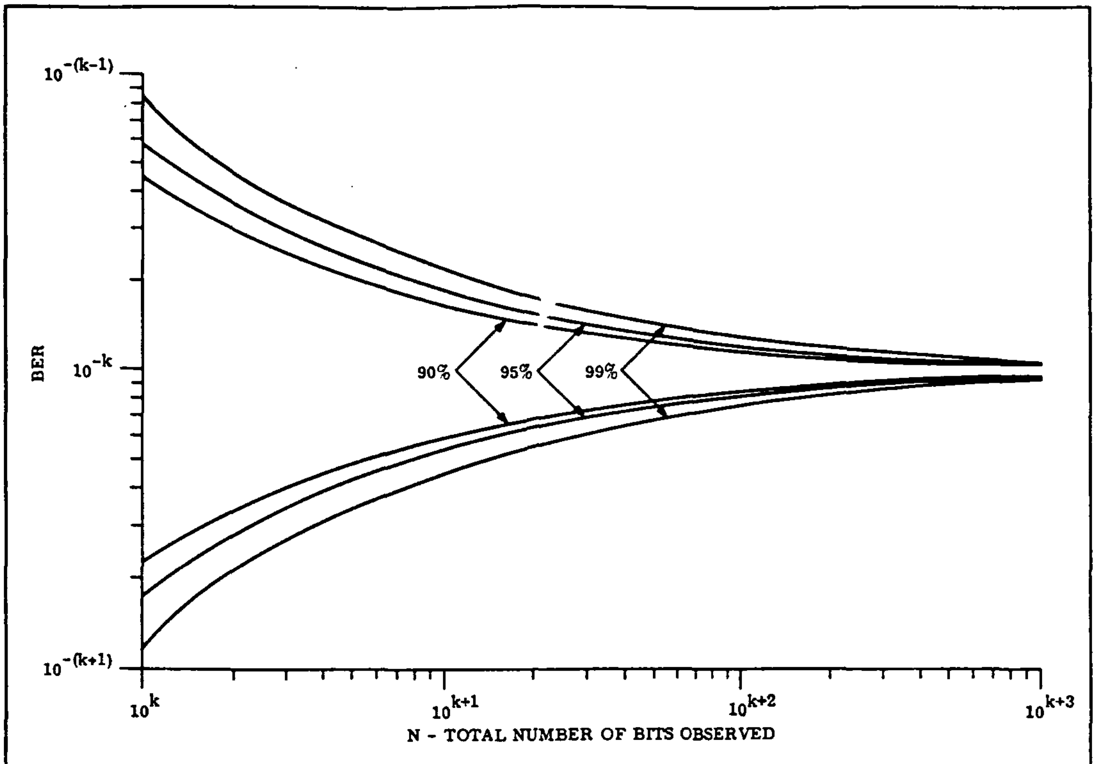

Figure 8. Confidence Bands When Observed Value is 10^{-k} [3]

If no assumptions are made about the distribution determining the $E[\hat{P}_e]$ or the variance of $\hat{P}_e$, use of the Chebyshev inequality is required to calculate a upper bound on the probability. The Chebyshev inequality has the form given by [13] as

$$P\{|\hat{P}_e - \mu| \geq t\sigma_{\hat{P}_e}\} \leq \frac{1}{t^2} \tag{21}$$

where

$\hat{P}_e$ is a random variable

$\sigma_{\hat{P}_e}$ is the standard deviation of $\hat{P}_e$

μ is the mean = $E[\hat{P}_e]$

t is the number of standard deviation units away from μ

For t = 10, the value of a particular $\hat{P}_e$ run will be within 10 standard deviation units of $E[\hat{P}_e]$.

3.3.3 Effect Of Dependent Errors. The assumption that error events are independent does not always hold true. A common example is differential encoding of the bit stream. The decoding process will tend to produce errors in pairs. According to [11], the variance of the estimator will be stretched by a factor of (1 + 2m) where errors occur in bursts of (1 + m). For differential encoding, m = 1 and the standard deviation of $\hat{P}_e$ will be multiplied by $\sqrt{3}$. Thus ε will be scaled by $\sqrt{3}$ for differential encoding. Also, the confidence interval will not be as tight as expected for the case of independent errors in a Monte Carlo simulation.

3.4 Summary

The theoretical curves of P_b vs E_b/N_o were plotted from Equation 5 and Equation 9 for ideal recovery of BPSK and DEBPSK respectively. Next, the code gain expected from the convolutional code was presented and plotted for the DEBPSK case. Code gains of from 3.8 dB to 5.8 dB are predicted for P_b ranging from 10^{-3} to 10^{-7}. Finally, the Monte Carlo P_b estimation technique was discussed. The error counting done in the simulation will be used to implement the Monte Carlo technique.

With the number of bits observed on the order of $10/P_b$, a 90% confidence interval is achieved for independent errors.

IV. BPSK Matched Filter Test

As a starting point for working with SPW™, a BPSK system was assembled and tested. An important part of this simulation is verification of the SPW™ noise generator block operation. Next, the configuration of the bit error counter is given. Also, timing considerations for the simulation are explained. Then, the testing process used is covered. Finally, the simulation results are presented and compared to the theoretical P_b.

4.1 BPSK Modulation in SPW™

A simplified block diagram of the simulated BPSK system is shown in Figure 9. The SPW™ block diagram is shown in Figure 17 in Appendix A. The SPW™ BPSK modulator block generates a constellation that is tilted 45° to (1, 1) and (-1, -1). The use of the complex tone generator block and mixer block internal to the BPSK modulation block forces energy onto both the in-phase and quadrature parts of the complex signal. When the carrier frequency is set to other than baseband, the output of the BPSK modulation block was found to have ±1 data changes on both channels, or a 45° tilt, instead of ±1 data on the in-phase channel and zero on the quadrature channel.

As developed in [14], the variance of the noise σ_n^2 depends on the value of η_o and the sampling rate f_{sa} of the simulation. The simulation bandwidth is shown in Figure 10. The equation for simulation bandwidth is

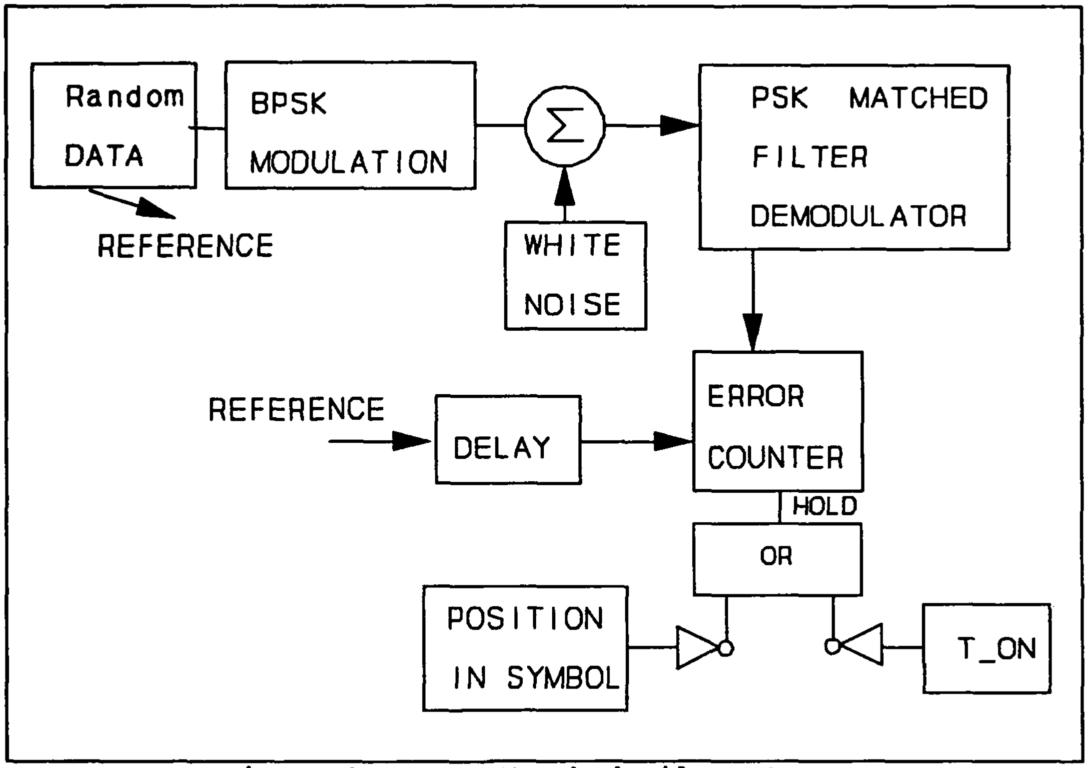

Figure 9. BPSK Matched Filter System

$$B_s = 2\left(\frac{f_{sa}}{2}\right) = f_{sa} \tag{22}$$

where f_{sa} is the sampling frequency.

The variance σ_N^2, or power in a zero-mean, real-valued Gaussian random process within a simulation bandwidth B_s is calculated to be

$$N_o = \sigma_N^2 = \int_{-\frac{f_{sa}}{2}}^{\frac{f_{sa}}{2}} S_N(f)\,df = \int_{-\frac{f_{sa}}{2}}^{\frac{f_{sa}}{2}} \left(\frac{\eta_o}{2}\right)df = \frac{\eta_o f_{sa}}{2} \tag{23}$$

where η_o is the white noise power spectral density.

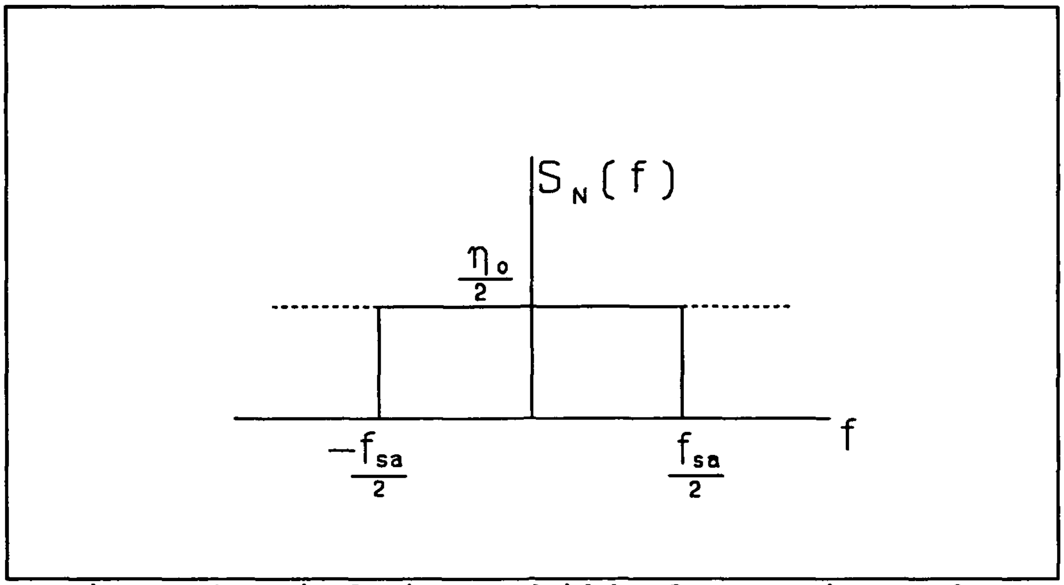

Figure 10. Simulation Bandwidth of a Gaussian Random Process

The SPW™ block white noise given signal-to-noise ratio and single-sided bandwidth produces complex Gaussian white noise samples. The noise variance is calculated from the following equation [10]:

$$\text{noise-variance} = (\text{signal\_average\_power}/\text{SNR})*(f_{sa}/(4*\text{BW})) \quad (24)$$

where

SNR is defined as $\frac{E_s}{N_o}$ with E_s the energy per symbol

BW is the symbol rate

The SPW™ equation, with some manipulation and noting that two times the single sided noise_BW gives the 2-sided noise bandwidth, is the same as Equation 23.

4.2 Data Flow In The System

The random data block produces random bits, 0 for logic low and 1 for logic high, at the system bit rate. The BPSK modulation block produces a baseband complex signal because the carrier parameter was set to 0 Hz. The 0 to 1 logic input is changed to an antipodal ±1 V bit stream internally before BPSK modulation. Next, a discussion of noise generation is required.

The SPW™ noise generation block detail is shown in Figure 18 in Appendix A. It takes the output from the BPSK modulation block as an input. The desired SNR is passed to the noise generation block as a parameter with units of decibel. The white noise given SNR and bandwidth block output starts as a zero mean and unit variance complex white noise block output. One block input and three parameters are used in the block calculations of scale factors required to produce the desired noise variance as given in Equation 24. The complex average power of the noiseless signal input to the block is calculated and divided by the SNR parameter. The sampling frequency and single-sided noise bandwidth parameters are used to generate the second scale factor. The complex output from the block is the Gaussian white noise with zero mean and variance adjusted to provide the desired SNR.

For the PSK matched filter system, the SNR specified for the white noise block is the E_b/N_o used to calculate the theoretical P_b. The noise is then summed with the complex BPSK signal. This step simulates AWGN of the channel. The noisy BPSK signal is sent to the SPW™ block PSK matched filter demodulator. A detail of the SPW™ PSK matched filter demodulator block is shown in Figure 19 in Appendix A. This block assumes perfect symbol and carrier synchronization. The constellation first angle is input as a parameter. The constellation first angle was set to 45°. The PSK matched filter demodulation block implements the matched filters as integrate-and-dump blocks. The block integrates for the bit duration calculated from the parameter bit rate. The output from the integrate-and-dump blocks is sent to a PSK detector block. The PSK detector block will compensate for the amount of channel phase rotation specified by the parameter channel phase rotation. The unrotated PSK signal is then sent to the PSK quantize block internal to the PSK detector block.

The PSK quantize block, Figure 20 Appendix A, computes the phase of the complex PSK signal at the end of the bit time. The calculated phase is then compared to the reference established internally by the parameters PSK modulation order and constellation first angle. The output of this block is the decoded complex bit stream. The real part of this complex signal goes to the modified real error rate counter block.

The modifications to the real error rate counter were made to generate the desired output of number of errors counted and number of bits counted. The value of initial error count was changed from 0 to 1 because it was found that the error count was one low with the first error event generating a value of 0 as configured by SPW™. The number of bits counted is available in the block, but not written as an output. The write block was moved from the probability of bit error location to the desired number of bits counted. The modified block was saved, and a new symbol generated for use in other systems.

4.3 Timing Considerations

In order to correctly compare the transmitted bits to the recovered bits, the transmitted bits are delayed by 10 iterations before entering the bit error rate counter. The 10 iterations is the time it takes for the integrate and dump block to produce the first valid data. The block parameter represents the block length. The error counter writes a result at the end of each block. The hold input on the error counter block was used to set the location in the bit of the error check. Both the transmitted and recovered bit streams are ±1 V square waves entering the block. The midpoint of the bit was chosen as the time to make the comparison. The T_ON block produces a logic false until the simulation time exceeds the parameter T_ON. The inverted false is a logic true input preventing the block from comparing data before valid data is available. After the simulation time exceeds T_ON, the T_ON block produces a logic true which is inverted to a logic low, allowing the position in symbol block to take over timing. The position in symbol block produces a logic true at the specified time in the symbol. The block parameters baud rate and symbol fraction are used to generate the correct timing. The position in symbol is set to 0.5, corresponding to the center of the bit.

4.4 Testing Process

The BPSK matched filter detection system shown in Figure 17 in Appendix A was tested for three values of SNR (4, 6, 8 dB). For each value of SNR, the theoretical probability of bit error P_b was calculated using Equation 5. As required by the Monte Carlo test method, the number bits run in the simulation was $10/P_b$. With ten samples per bit, the required number of iterations was $100/P_b$, with an additional 10 iterations to fill the matched filter. For the simulation, the arbitrary bit rate was 5 bps and the sampling frequency was 50 Hz. This maintains the desired 10 samples per bit. The ratio sample rate to bit rate is the important feature, not the individual values [2]. The carrier frequency was 0 Hz. The initial error count was 1. T_ON was 0.2. Table 2 summarizes the parameters and the parameter values.

Table 2. Test Parameters

Parameter	Name	Value
sample frequency	Sfreq	50 Hz
bit rate uncoded	bitrate	5 bps
number of bits to count	block	1,000
signal-to-noise ratio	SNR	3.5 dB
first error event	initial_error	1
time to start error count	T_on	0.2 sec

Table 3 shows the three SNR values, block lengths, and theoretical probabilities of bit errors.

Table 3. Theoretical P_b

SNR	Block Length	Number of Iterations	Theoretical P_b
4.0 dB	1,000	10,010	$1.25008*10^{-2}$
6.0 dB	10,000	100,010	$2.38829*10^{-3}$
8.0 dB	100,000	1,000,010	$1.90908*10^{-4}$

For each simulation, the number of iterations was (10 * block length) + 10. The noise seeds were arbitrarily chosen. The ten noise seeds are shown in Table 4.

Table 4. Noise Seeds

Noise Seed	Simulation Run
5,000	1
10,000	2
1,500	3
60,000	4
70,000	5
75,000	6
100,000	7
200,000	8
20,000	9
35,000	10

4.5 Test Results BPSK Matched Filter System

Table 23 through Table 26 in Appendix B show the results of the testing. Table 5 shows the statistics of the test results.

Table 5. $\hat{P}_b$ Statistics Matched Filter Detection BPSK

E_b/N_o	μ	σ^2	σ	ε
4.0 dB	1.34×10^{-2}	2.538×10^{-5}	5.038×10^{-3}	0.403
6.0 dB	2.84×10^{-3}	3.804×10^{-7}	6.168×10^{-4}	0.258
8.0 dB	2.43×10^{-4}	3.157×10^{-9}	5.618×10^{-5}	0.294

The estimation error ε is less than 1 for all three levels of SNR. This is an acceptable indication of the goodness of the estimator [12]. The mean value of the ten simulation runs was plotted for each E_b/N_o level. Additionally, vertical lines representing the value of $\pm\sigma$ at each E_b/N_o level were plotted. The curve labeled measured data in Figure 11 is within one standard deviation of the theoretical curve.

Although investigation of simulation time was not a stated goal, it did become important at higher E_b/N_o levels. Each noise seed simulation took approximately 10 minutes at the 4 dB E_b/N_o level. As a comparison, over one and a half hours were required to simulate 10^5 bits with 1.0001×10^6 iterations in the interactive mode of the Block Diagram Editor (BDE)

4.6 Summary

The simulation of BPSK matched filter detection accomplished three things. First, operation of the bit error counter was verified. Second, the operation of the white noise given SNR and bandwidth was confirmed. The verification of correct operation is made by comparison of the P_b vs E_b/N_o curve resulting from the simulation to the theoretical curve. The P_b curve generated from

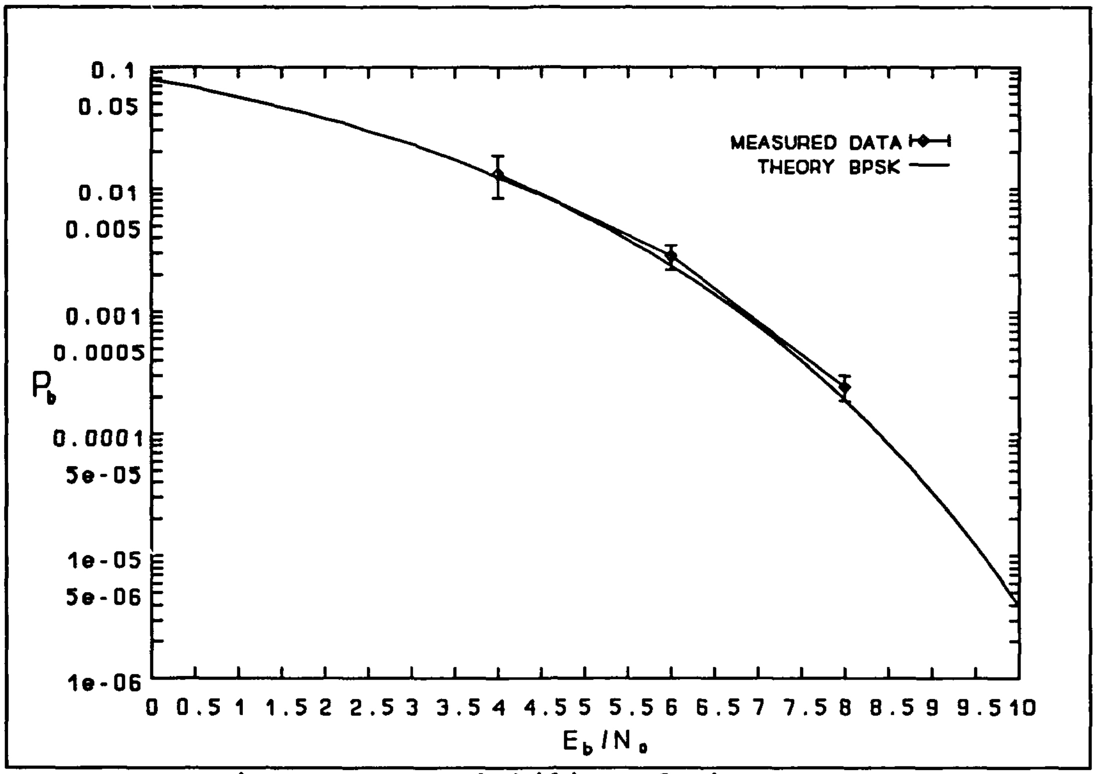

Figure 11. Probability of Bit Error

the measured data was within one standard deviation of theoretical. Finally, the estimator error ε was shown to be less than one.

V. Costas Loop

This chapter begins with a discussion of the use of the Costas Loop as a demodulator. Next, the mathematical equations defining Costas loop operation are covered. Then, the equations necessary for calculation of simulation parameters are presented. The theoretical portion of this chapter ends with a table of parameter values initially used for the simulations.

In the second half of this chapter, the SPW™ Costas loop block operation is covered. The simulation test systems used to find the Costas loop response to a phase or frequency step are discussed. Next, the SPW™ Costas loop block modifications are discussed. Finally, the modified Costas loop simulation testing is presented and the results compared to theory.

5.1 Costas Loop Demodulation Theory

The Costas loop or, I-Q loop, is a method of suppressed carrier recovery. After initial carrier acquisition, the Costas loop provides a demodulated bit stream on the I arm, as shown in Figure 12. The noise spectrum shaping of the Costas loop arm filters can be designed to be the same as the low-pass equivalent of the input filter $H_i(S)$ shown in Figure 13. When this condition is met, the performance of the Costas loop is identical to the squaring loop [15]. The squaring loop removes the modulation by squaring the received signal. The phase-locked loop (PLL) then tracks the double-frequency term from the squarer. After being divided by two, the recovered carrier is

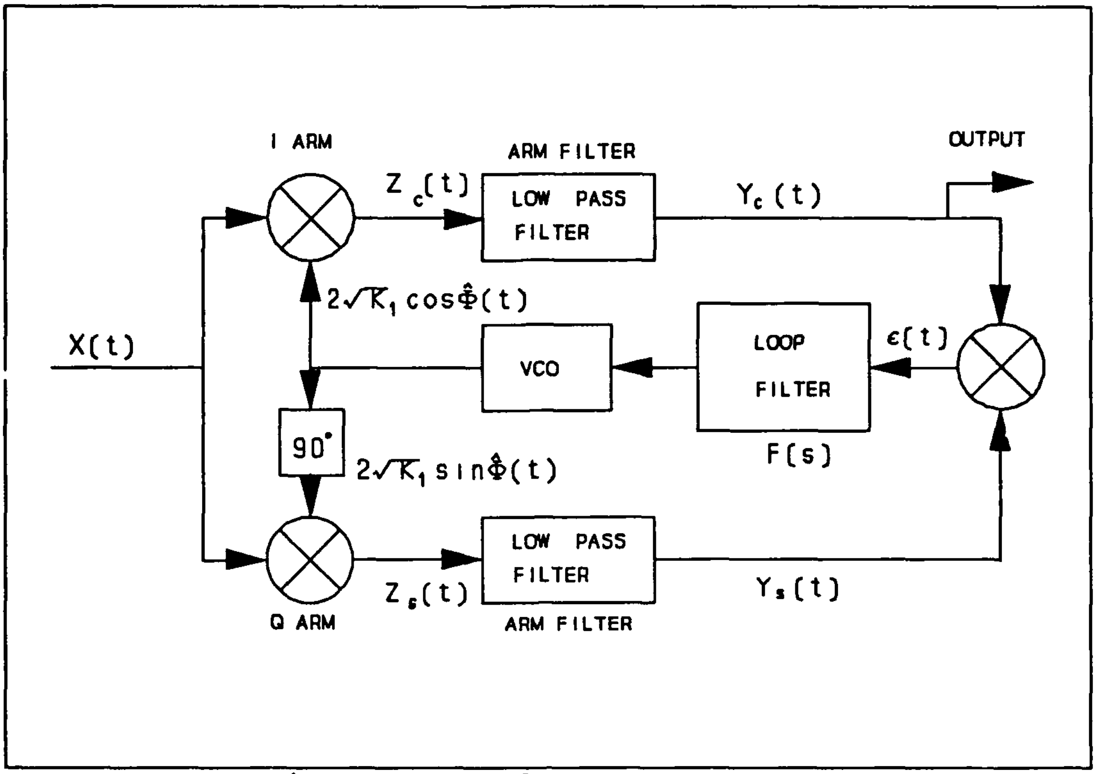

Figure 12. Reference Costs Loop

available as a coherent reference for demodulation.

As developed in [15], the Costas loop with multiplier gain $\sqrt{K_m} = 1$ and with signals labeled as in Figure 12 has an input given by

$$X(t) = \sqrt{2S}m(t)\sin\Phi(t) + n(t) \tag{25}$$

where

$\Phi(t) = \omega_o t + \theta(t)$

$m(t)$ is the signal modulation

$n(t)$ is the additive noise

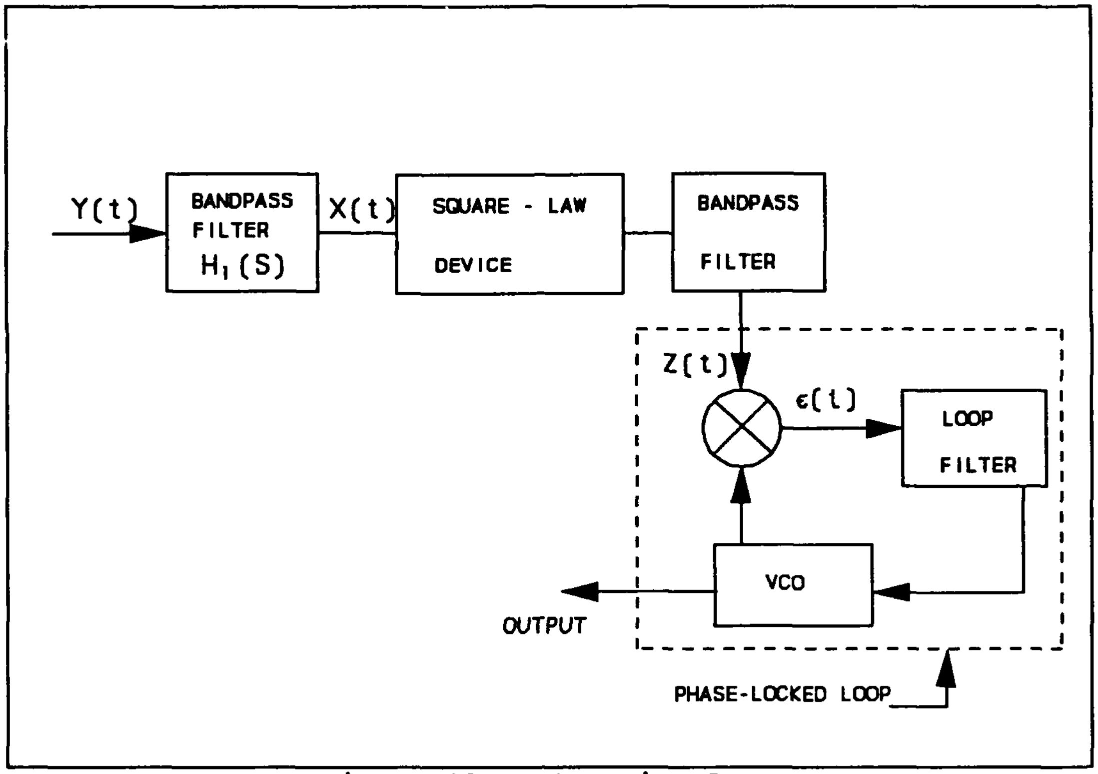

Figure 13. Squaring Loop

This received signal is mixed with the output of the voltage controlled oscillator (VCO). The resulting in-phase signal is given as

$$Z_c(t) = X(t)\left[2\left(\sqrt{K_1}\right)\cos\hat{\Phi}(t)\right] \tag{26}$$

where

$\hat{\Phi}(t) = \omega_o t + \theta(t)$ is the estimate of $\Phi(t)$

$k(t)$ is the VCO output power

Substituting Equation 25 into Equation 26 and filtering the double frequency terms yields:

$$Y_c(t) = \sqrt{2K_1}\left[\sqrt{S}m(t) - N_s(t)\right]\sin\phi(t) + \sqrt{2K_1}N_c(t)\cos\phi(t) \tag{27}$$

Similarly, for $X(t)$ mixed with the quadrature part of the VCO output, $Z_s(t)$ yields:

$$Y_s(t) = \sqrt{2K_1}\left[\sqrt{S}m(t) - N_s(t)\right]\sin\phi(t) - \sqrt{2K_1}N_c(t)\cos\phi(t) \tag{28}$$

where

$\phi(t) \doteq \theta - \hat{\theta}$ is the phase error estimate

The phase detector mixer multiplies the $Y_s(t)$ and $Y_c(t)$ outputs from the arm filters, producing the phase error signal $\varepsilon(t)$.

$$\varepsilon(t) = k_1\left\{\left[\sqrt{S}m(t) - N_s(t)\right]^2 - N_c^2(t)\right\}\sin 2\phi(t) + 2k_1N_c(t)\left[\sqrt{S}m(t) - N_S(t)\right]\cos 2\phi(t) \tag{29}$$

Finally, the loop filter closes the tracking loop by providing a filtered version of the $\varepsilon(t)$ as the control voltage input to the VCO.

Four things are noteworthy in the previous equations. For digital modulation, m(t) is a ±1 pulse train; therefore, $m^2(t) = 1$. The loop is operating on the square of the noise. Also, twice the phase error is being tracked. Finally the magnitude of $\varepsilon(t)$ depends on the power of the received signal S. The automatic gain control in an actual receiver keeps S constant. The 2ϕ and squared noise term introduce a squaring loss term into the loop bandwidth equation. The loop noise bandwidth and loop filter transfer function F(s) determine the performance of the Costas loop. Because the heart of the Costas loop is a PLL, equations for a PLL were used to determine parameters such as VCO gain and loop filter natural frequency in the model.

Reference [16] documents a practical method of designing a PLL once the desired lock range is chosen. For the model, the known tracking parameter was the pull-out range ω_{po}. The pull-out range is the dynamic limit of stable operation once lock has been established. As defined in [16]

$$\Delta\omega_{po} \approx 1.8\omega_n(\zeta + 1) \tag{30}$$

where

ω_n is the natural frequency of the PLL

ζ is the damping factor

The lock-in range is defined as [16]

$$\Delta\omega_L = 2\zeta\omega_n \tag{31}$$

Given $\Delta\omega_{po}$ and ζ, the loop natural frequency can be calculated using Equation 30. Once ω_n is known, a value for $\Delta\omega_L$ can be calculated using Equation 31.

Reference [16] assumes the values of K_o and K_d will be found in a specification for the PLL device. As was done by [16], for simulation on a computer:

$$K_oK_d = 10\ \omega_n \tag{32}$$

where

K_o is the VCO gain constant

K_d is the phase detector gain

This relation was initially used to meet the requirement that $K_oK_d \gg \omega_n$. Assuming K_d in the simulation is 1, a value for K_o can be calculated from Equation 32. The modem design expects $\Delta f_{po} = 2\pi\Delta\omega_{po}$ to be 20 Hz or less [6]. Equation 30 and Equation 31 were used to calculate theoretical values of f_n = 6.5087382 Hz and Δf_L = 9.204 Hz respectively. Equation 32 yields a theoretical K_o of 65.087382 Hz/V. The loop filter characteristics, VCO gain K_o, and phase detector gain K_d determine ω_n. To model the modem loop filter, a passive lead-lag filter is used. For $\tau_1 \gg \tau_2$ and $\tau_2 \gg 1/K_oK_d$, the passive filter approaches the perfect integrator form of a proportional plus integrator loop filter [7].

A passive lead-lag loop filter has the transfer function given by [7] as

$$F(s) = \frac{s\tau_2 + 1}{s\tau_1 + 1} \tag{33}$$

where τ_1 and τ_2 are the time constants of the filter.

For a passive loop filter of the form in Equation 33, [16] gives the equations for ω_n and ζ as

$$\omega_n = \left(\frac{K_o K_d}{\tau_1 + \tau_2}\right)^{\frac{1}{2}} \tag{34}$$

$$\zeta = \frac{1}{2}\left(\frac{K_o K_d}{\tau_1 + \tau_2}\right)^{\frac{1}{2}} \left(\tau_2 + \frac{1}{K_o K_d}\right) \tag{35}$$

The term K_oK_d is called the loop gain. According to [16], the loop is a high gain loop if $K_oK_d >> \omega_n$. If a PLL with a passive loop filter is a high-gain loop, the phase transfer function can be approximated as [7,16]

$$H(s) \approx \frac{2s\zeta\omega_n + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{36}$$

This is the same transfer function obtained if the PLL loop filter is an ideal proportional plus integrator filter [16].

As was done by [17], the Costas loop model operation was verified using a noiseless simulation. The transient phase error to step changes in phase and frequency were the test parameters. For a step in phase, [7,17] predicted a phase error response where time to peak deviation was used as the criteria for proper performance. For a step in frequency, the time for the loop to lock and steady-state phase error were used as the criteria for correct performance. The phase step must be small enough for the loop to remain in the linear region of operation [7]. The small angle approximation $\sin\theta \approx \theta$ for $\theta \leq 30°$ is commonly used to determine the linear range. The time to the peak phase error for the Costas loop using a second-order loop with $\zeta = 0.707$, a phase step of 20°, and $f_n = 6.5087$ Hz is 0.0543 sec. For testing the loop response to a frequency step, the frequency step must be less than or equal to $\Delta\omega_L$. If the frequency step exceeds $\Delta\omega_L$, the loop will unlock and a pull-in process will occur instead of the desired lock-in process. The time to pull in of a PLL is typically 1000 times larger than the time to lock [16]. The time for a PLL to lock (T_L) onto a step in frequency, according to [7], can be approximated by

$$T_L \approx \frac{1}{\omega_n} \tag{37}$$

For $f_n = 6.5087$ Hz, $T_L \approx 0.1536$ sec. The steady-state phase error for a second-order loop using a passive loop filter is given as

$$\theta_e(\infty) = \frac{\Delta\omega}{K_o K_d} \tag{38}$$

where $\Delta\omega$ is the frequency step in radians.

For the simulation with $\Delta\omega = \Delta\omega_L = 2\pi\Delta f_L = 9.2047$ Hz and $K_o K_d =$ 65.087 Hz/V, $\theta_e(\infty) = 0.1414$ radian. Equation 34 for ω_n and Equation 35 for ζ were solved for expressions for τ_1 and τ_2. The resulting expressions are

$$\tau_1 = \frac{K_o K_d}{\omega_n^2} - \tau_2 \qquad (39)$$

$$\tau_2 = \frac{2\zeta}{\omega_n} - \frac{1}{K_o K_d} \qquad (40)$$

The noise bandwidth and signal-to-noise ratio of the loop were the last considerations of the theoretical operation of the loop. In a Costas loop with two-pole Butterworth arm filters, the squaring loss S_L is approximately -3 dB [9]. From [9] the equivalent loop signal-to-noise ratio is

$$\rho_l = \frac{1}{4\sigma_\phi^2} \qquad (41)$$

with

$$\sigma_\phi^2 = \frac{1}{\rho S_L} \qquad (42)$$

$$\rho = SNR_i\left(\frac{B_i}{2B_L}\right) \qquad (43)$$

where

SNR_i is the signal-to-noise ratio into the loop

B_i is the arm filter bandwidth

B_L is the single-sided loop noise bandwidth of the linear loop, given by:

$$B_L \approx \frac{1}{2}\left(\frac{2\zeta^2 + 1}{2\tau_2}\right) \tag{44}$$

Examination of Equation 41 for ρ_1 with substitution of variables by Equation 42 and Equation 43 reveals that it changes by an amount directly proportional to SNR_i. Therefore, a constant ρ_1 can not be maintained for the Costas loop suppressed carrier recovery loop [9]. For the theoretical values of τ_2 and ζ given in Table 6, B_L = 14.9887 Hz. The arm filter bandwidth = 9600 Hz, and SNR_i ranges from 3.5 dB to 5.5 dB. The theoretical range for ρ_1 is approximately 19.5 dB to 21.5 dB.

5.2 Summary of Theoretical Operation Costas Loop

Working with the assumption that the modem acquisition circuits will perform the pull-in operation, the pull-out range for a locked loop was the starting point for the calculation of loop parameters. The parameters of interest for the simulation were loop damping ζ, loop natural frequency f_n, and VCO gain K_o. Given Δf_{po} = 20 Hz and $\zeta \approx 0.707$, Table 6 shows the theoretical values for loop operation.

5.3 The SPW™ Costas Loop Block

The Costas loop block as implemented by the SPW™ software requires the user to specify seven parameters. Figure 21 in Appendix A is the unmodified Costas loop block. Sampling frequency, natural frequency in Hz, damping factor, and VCO

Table 6. Theoretical Costas Loop Parameter Values

Parameter	Value
ζ	$\sqrt{2}/2$
Δf_{po}	20 Hz
f_n	6.5087382 Hz
T_L	0.1536396 sec
Δf_L	9.2047 Hz
K_oK_d	65.087382 Hz/V
τ_1	0.2123892 sec
τ_2	0.0321358 sec

constant in Hz/V are all used to control the operation of the loop filter. The loop filter is a passive lead-lag filter. The VCO quiescent frequency and the VCO gain are passed to VCO block and used to control its operation. The arm filter order and arm filter bandwidth in Hz are passed to the complex Butterworth lowpass infinite impulse response filter block. A complex multiplier block is used to mix the complex input. The gain of this block is unity. After the complex output from the Butterworth filter is separated into real and imaginary parts, a real unity gain multiplier performs the phase detection process. The complex conjugate bock produces the desired $e^{-j\theta}$ signal from the VCO output. The $e^{-j\theta}$ multiplied by $e^{j\theta}$ input removes the channel rotation shown in Equation 2. Finally, the one iteration hold block between the lead-lag filter and the VCO allows time for calculations and outputs to be generated in the feedback path.

The first modification made to the Costas loop block was the addition of output ports for the in-phase and quadrature parts of the Butterworth filtered signal. These outputs correspond to the real-arm and imaginary-arm signals, as shown in Figure 22 in Appendix A. Also added was an output port allowing access to the phase error from the phase detector. The block with the above modifications was saved and a new symbol created to represent the new block as shown in Figure 23 in Appendix A.

5.3.1 Costas Loop Testing. For testing the Costas loop, two systems were built. Figure 24 in Appendix A shows the phase step test system with final parameter values. The phase step system multiplies the BPSK signal by a constant $e^{j\theta}$ term. The frequency step test system simulates a frequency step by offsetting the VCO center frequency by an amount in Hertz equal to the desired frequency step. Figure 25 in Appendix A shows the frequency step test system. The operation and test results of these two systems are given in the following sections.

5.3.2 Phase Step Test System and Results. The BPSK signal to be multiplied by the $e^{j\theta}$ term was generated with the BPSK modulation block fed by a random data block. The carrier frequency in the BPSK block was set to 0 Hz, and the bit rate of the random data block was set to 2400 bps. Using the rule of 10 samples per bit in the simulation, the sample frequency was set to 24000 Hz. The constant block value was also set to 24000 and connected to the signal sink blocks sampling frequency input (s_freq). The arm filter bandwidth of the two-pole Butterworth filter in the Costas loop was set to 4800 Hz. Initial testing was done with the theoretical values for loop frequency and VCO gain. The parameters were set to 6.5087 Hz and 65.087 Hz/v. Table 7 summarizes parameter settings for the test.

Table 7. Phase or Frequency Step Test Parameters

Parameter	Name	Value
sample frequency	sfreq	24000 Hz
bit rate	bitrate	2400 bps
loop frequency	loopfreq	6.5087382.0 Hz
VCO gain	VCO_gain	65.087382.0 Hz/V
arm filter bandwidth	arm_band	4800.0 Hz
step for phase test	value	20.0°
step for frequency test	VCO_FREQ	9.2047 Hz

The phase step θ in radians was produced by a constant block with value set equal to the phase step in degrees fed through a conversion block degree to radian. The radian value of the phase step was fed as the x input to an e^{jx} block thus producing the desired $e^{j\theta}$ term. The VCO output of the Costas loop is taken before the complex conjugate block as shown in Figure 22 in Appendix A. The complex phase block converts the VCO output to a phase. Correct Costas loop operation will produce a signal stored in the signal sink labeled billsigs/phase_out that after a transient response matches the phase step. Table 8 lists other signal sink block names and what they recorded.

Correct operation of the Costas loop would be verified by the phase error transient and steady-state response. The stored signals phase_in and phase_out were used to calculate the

Table 8. Phase Step Test Signal Sinks

Sink Name	Contents
bpsk_out	modulated signal
phase_in	phase step signal in radians
I_ARM	in-phase arm of Costas loop after Butterworth filter
Q_ARM	quadrature arm of Costas loop after Butterworth filter
phase_error	phase error from phase detector
VCO_control	Lead-Lag filter output
VCO_out	voltage controlled oscillator output
phase_out	phase of voltage controlled oscillator output

normalized phase error. The math functions in the Single Display Editor part of SPW™ were used to perform the required calculations on the signals.

The first test runs of the test system failed to produce the expected rise time to peak phase error. The phase step size was varied over a range of 5° to 20° with little change in the loop response. The phase of the VCO output would slowly rise to a peak and oscillate until eventually settling to the phase step value. This response was similar to loops with ζ values of 0.3 or less as shown by [7,16].

5.3.3 Frequency Step Test System and Results. The system used for testing the Costas loop response to a step in frequency is shown in Figure 25 in Appendix A. The BPSK modulated signal was generated as a baseband signal. The frequency step was modeled as a frequency off-set of the VCO quiescent frequency. The parameter VCO_FREQ was used to set the VCO frequency to the required step size. The complex phase block was used to calculate the phase of the Costas loop VCO output during the lock-in response. Other system blocks operate as described in Section 5.3.2 on the phase step test system. Table 7 summarizes parameter settings for the test.

The complex tone block was set to generate a tone at the theoretical loop natural frequency f_n. Comparison of this signal to the VCO output signal demonstrates the lock-in within one cycle of f_n. Table 9 lists the signal sink names and what was stored in them.

Correct operation of the Costas loop would be verified by the time to lock and steady-state phase error value. The test system runs failed to produce the expected results. For a range of frequency steps of 3 Hz to 9.2047 Hz, the Costas loop appeared to perform a pull-in process over time periods much larger than T_L. The description given in [16] of a pull-in process matched the Costas loop response. However, with the $\Delta\omega$ less than $\Delta\omega_L$, a lock-in process should have been observed.

5.4 Debugging the Costas Loop.

The theoretical calculations were checked, and the test system construction was thoroughly checked. With the knowledge that the Costas loop internal parameter τ_1 and τ_2 were being calculated by the software, the equations used for the calculations were found and were

Table 9. Frequency Step Test Signal Sinks

Sink Name	Contents
bpsk_out	modulated signal
loop_freq	tone generator output at f_n
I_ARM	in-phase arm of Costas loop after Butterworth filter
Q_ARM	quadrature arm of Costas loop after Butterworth filter
phase_error	phase error from phase detector
VCO_control	Lead-Lag filter output
VCO_out	voltage controlled oscillator output
phase_out	phase of voltage controlled oscillator output

$$\tau_1 = \frac{VCO\_constant}{\left(\frac{\pi}{5}\right) f_n^2} \tag{45}$$

$$\tau_2 = \frac{Damping\_factor}{\pi f_n} - \frac{1}{2\pi VCO\_constant} \tag{46}$$

where

VCO_constant is the VCO gain in Hz/V

damping_factor is the loop filter damping

f_n is the loop natural frequency in Hz

Comparison of these equations to the theoretical Equation 39 and Equation 40 as presently written is difficult. Rewriting Equation 39 and Equation 40 to change the units from radian to hertz produces

$$\tau_1 = \frac{K_o K_d}{2\pi f_n^2} - \frac{\zeta}{\pi f_n} + \frac{1}{2\pi K_o k_d} \quad (47)$$

$$\tau_2 = \frac{2\zeta}{2\pi f_n} - \frac{1}{2\pi K_o K_d} \quad (48)$$

where

f_n is the loop natural frequency in Hz

K_oK_d is the product of the VCO gain and detector gain in Hz/V

ζ is the loop filter damping

With the assumption that the detector gain K_d is unity, Equations 46 and 48 for τ_2 are identical. Equations 45 and 47 for τ_1 are noticeably different. Even with the assumption that $\tau_1 >> \tau_2$, allowing the last two terms of Equation 47 to be neglected, the remaining equations do not agree. As a test, the theoretical value was written in as the parameter value for τ_1 in the lead-lag filter block. This effectively deleted the software equation that had been calculating τ_1. The Costas loop response to the phase step with the theoretical value of τ_1 in place of the software calculated value agreed with the theoretical rise time.

To correct the Costas loop equation used to calculate τ_1, the expression for τ_1 in the lead-lag filter was modified to implement Equation 47. The modified expression for the τ_1 parameter was

$$\tau_1 = \frac{VCO\_constant}{2\pi f_n^2} - \frac{damping\ factor}{\pi f_n} + \frac{1}{2\pi VCO\_constant} \quad (49)$$

where

VCO_constant is the VCO gain in Hz/V

f_n is the loop natural frequency in Hz

The modified Costas loop was saved as a new block. The phase and frequency step test systems were updated to use the modified Costas loop and testing of the loop completed.

5.5 Summary of Costas Loop Testing

5.5.1 Phase Step. For a 20° phase step with the theoretical values for the loop parameters, Table 10 shows the test results. Representative outputs of correct operation are shown in Appendix A, Figures 26 - 29.

Table 10. Costas Loop Test Result For VCO Gain 65.087382 Hz/V and a Phase Step Input

Parameter	Theoretical	Measured
rise time	0.0543 sec	0.0544 sec
normalized peak deviation	-0.208	-0.167
steady_state error	0.0°	0.0°

The assumption of a high gain loop in [7] did not give a value for how much greater K_oK_d must be than ω_n. For the data presented by [16], $K_oK_d = 10\omega_n$ was used. This also was used for the theoretical calculations used to calculate the VCO gain parameter resulting in Table 10 results. As an additional test, the simulation was repeated for the same parameters except with double the VCO gain. Table 11 shows the results of using 130.17476 Hz/V for the VCO gain.

Table 11. Costas Loop Test Result For VCO Gain 130.17476 Hz/V and a Phase Step Input

Parameter	Theoretical	Measured
rise time	0.0543 sec	0.0559 sec
normalized peak deviation	-0.208	-0.192
steady_state error	0.0°	0.0°

Rise time changed from within 0.18% to within 2.9% of theoretical. Peak deviation changed from within 19.7% to within 7.7% of theoretical. Based on the above test results, the VCO gain used for future simulations was set to 130.17476 Hz/V.

5.5.2 Frequency Step. The results of the testing the frequency step test system are shown in Table 12. Testing was done with parameter values given in Table 7 with the exception of the VCO gain which was set to 130.17476 Hz/V. Typical outputs of correct operation are shown in Figures 30 - 33 in Appendix A.

Table 12. Costas Loop Frequency Step Test Result

Parameter	Theoretical	Measured
time to lock	0.1536 sec	0.127 sec
steady_state phase error	0.0707 radian	0.0745 radian

The steady-state phase error is within 5.5% of theoretical. Figures 30 and 31 in Appendix A show the lock-in process. Figure 30 in Appendix A is the complex VCO output and a tone generator output at the loop natural frequency. The VCO center frequency was offset by 9.2047 Hz from the 0 Hz carrier. The VCO output clearly shifts to a +1 on the real and 0 on imaginary trace within one cycle of the signal representing the loop natural frequency. The VCO output then settles to the steady-state error value.

5.6 Summary

This chapter covered the theoretical development of the Costas loop parameters. The SPW™ block was tested with the theoretical parameter values, and retested with modifications to the parameter equations used to calculate τ_1 and τ_2. The Costas loop with the modified equation for τ_1 produced a simulated response to a phase step that was within 2.9% of the theoretical rise time and within 7.7% of the theoretical peak deviation. The modified loop produced a simulated response to a frequency step that was within 5.5% of the theoretical steady-state phase error and lock-in was within one cycle of the loop natural frequency, as predicted by theory. The VCO gain parameter value was doubled from 65.087382.0 Hz/V to 130.17476 Hz/V. A trade was made of slower rise time for a better peak deviation for the phase step response. Rise time changed from within 0.18% to within 2.9% of theory, whereas peak deviation changed from within 19.7% to within 7.7% of theory. Without the discovery and correction of the problem with the SPW™ Costas loop equation for the τ_1 parameter, the simulation of the modem would not have worked properly.

VI. Coded BPSK System Simulation

This chapter explains the configuration of the SPW™ system used to model the coded BPSK system. First, the configuration and testing of the blocks used to encode the random data are explained. Then, a description of the completed system used in the simulation is given. Finally, the results of the simulation are presented.

6.1 Encoder and Decoder Block Descriptions

With the Costas loop modification and testing completed, two other systems remained to be implemented in SPW™. The differential encoder and decoder blocks were built and tested. Also, the convolutional encoder and Viterbi decoder blocks were configured to model the modem. Configuration and testing of the encoder and decoder blocks are the topics of the next sections.

6.1.1 Differential Encoder and Decoder. The differential encoder and decoder blocks were built from standard SPW™ blocks. Figure 34 and Figure 35 in Appendix A show the encoder system and symbol respectively. The encoder takes in 0 for a logic low and a 1 for a logic high. The delay block parameter is set to the number of samples in a bit. The block was tested by an input of a know bit stream producing a correctly coded output. The differential decoder system and symbol are shown in Figure 36 and Figure 37 in Appendix A. The input to this block must also be 0 for a logic low and 1 for a logic high. The delay parameter must be set to the number of samples per bit. The block operation was verified by its correct decoding of the differential encoder block output.

6.1.2 Convolutional Encoder Configuration. The SPW™ convolutional encoder block is well-documented in [10]. The block can be programmed to generate many different convolutional codes. The block has the following limits [10]:

K the number of data bits input at a time must be < 10

N the number of coded bits for each K bits must be < 10

L the code constraint length must be < 10

also

K * (L - 1) must be less than 15

Given the above limits the rate 1/2, L = 7 or 9 code of the modem can be modeled, but the rate 3/4, L = 7 or 9 codes can not be modeled. Another consideration for block configuration is the block requirement for

$$K * (\text{samples/bit in}) = N * (\text{samples/symbol out}) \qquad (50)$$

For the rate 1/2 code with 10 samples/bit in, the coded bits have only 5 samples/symbol. Care must be taken to ensure integer relations exist or the block will force the result to the nearest integer.

For timing purposes, it is noted that the first valid output from the block is delayed according to the equation [10]

$$((K - 1) * (f_{sa}/R_b)) + (\text{nearest integer}(\%T_s * f_{sa}/R_b)) - 1 \qquad (51)$$

where

f_{sa} is the sampling frequency

R_b is the input bit rate

$\%T_s$ is the fraction of symbol position: the parameter that determines where the input bit is sampled.

The clock output from the block goes logic high for the first sample of each output encoded bit after an initial delay given by Equation 51. Finally, a text file must be made to specify the generator polynomials and values of N, K, and L. Following the required format given in [10], with N = 2, K = 1, L = 7, and the modem generator polynomials, the text file code_gen was made.

Text File Code_gen

```
2 1 7
 171
 133
```

The 171 in the file is the octal representation of the G_0 generator polynomial 1111001 (binary). The 133 is the octal representation of the G_1 generator polynomial 1011011 (binary). In addition to the text file code_gen, three parameters must be set in the block. The sample frequency and uncoded bit rate are required. Finally, the position in the bit where the coder will take a sample is specified by the fraction of symbol position parameter.

6.1.3 Viterbi Decoder Configuration. The SPW™ Viterbi decoder using soft decisions is well-documented in [10]. The same text file used for configuration of the conventional encoder is used by the Viterbi decoder. The limits given in Section 6.1.2 on N, K, L, also apply to the decoder. An additional decoder limit is the truncation path length. The truncation path length is a Viterbi decoder parameter. It determines how much path history will be stored and used in the decoder decision process. The truncation path length limit is 100. The Viterbi decoder implements soft decisions by first quantizing the input data stream and then making decisions weighted by the confidence metrics. A second text file is required by the decoder block. This text file provides information on the quantization threshold levels and metrics. For the modem model, eight-level quantization of the ±1 antipodal signal is used. As determined from [6], the modem threshold levels can be modeled as [-1, -0.66, -0.33, 0, 0.33, 0.66, 1]. The file named decode_file was created and has the form shown below.

Text File Decode_file

8	7	0
-1.00	6	1
-0.66	5	2
-0.33	4	3
0.00	3	4
0.33	2	5
0.66	1	6
1.00	0	7

The 8 in the file specifies the number of quantization levels, and the threshold levels are given directly below the 8. The two columns of numbers 7 through 0 and 0 through 7 are the confidence levels given to a received threshold level. The second column of numbers, 7 through 0, is the transition metric used when a given quantization level is received given that a logic low was transmitted. The third column of numbers, 0 through 7, is the transition metric used when a given quantization level is received given that a logic high was transmitted. For example, a quantization level 1 is lightly weighted as a logic low by the 0 in column three, and heavily weighted as a logic high by the 7 in column two. The output of the Viterbi decoder is a decision that a logic high or low was received based on the quantization and confidence levels applied to the block input. In addition to the two text files code_gen and decode_file, two parameters must be set. The sample frequency and encoded symbol rate must both be specified in the block.

For correct decoding, the block timing input must be logic high at the position in the encoded bit where the quantization is to occur and logic low at other times. The clock should be periodic at the encoded bit rate. The convolutional encoder block clock output is available for timing the decoder. Use of the encoder output clock for decoder timing makes the assumption of perfect symbol synchronization in the demodulation process. The clock from the encoder must be delayed by an amount equal to any system delay between the encoder output and the decoder input. With the encoded data input and the correct timing, the decoder will produce the original uncoded bit streams at the block output. There is a finite delay between the first encoded bit input to the decoder and the first uncoded output. The equation given in [10] for the delay is

$$(PL * (f_{sa}/R_s) * N) + (f_{sa}/R_s) * (N - 1) \tag{52}$$

where

f_{sa} is the sampling frequency

R_s is the input coded symbol rate

N is the number of coded bits produced for each uncoded bit input

PL is the path length used for Viterbi decoding

This delay becomes part of the system delay that must be accounted for when the original random data at the system input is used as a reference to check the system output.

<u>6.1.4</u> <u>Back-to-Back</u> <u>Testing.</u> The convolutional encoder with a random data input was connected directly to the Viterbi decoder in a noiseless environment. The encoder clock out was sent directly to the decoder clock in, and the inverted clock out was sent to the decoder hold. The decoder hold will suspend decoder operation between valid points in the bit stream. Signal sinks were used to record the random data input, encoded data, clock, and decoded data. A test of the system at an uncoded bit rate of 2400 bps and sample frequency of 24000 Hz produced no bit errors. The timing signal was verified by inspection of the recorded data as occurring at the first sample of the encoded bit.

6.2 Complete System Testing

After the various component blocks had been tested for operation, the complete model was assembled. Figure 14 is a simplified block diagram of the system. The actual system with

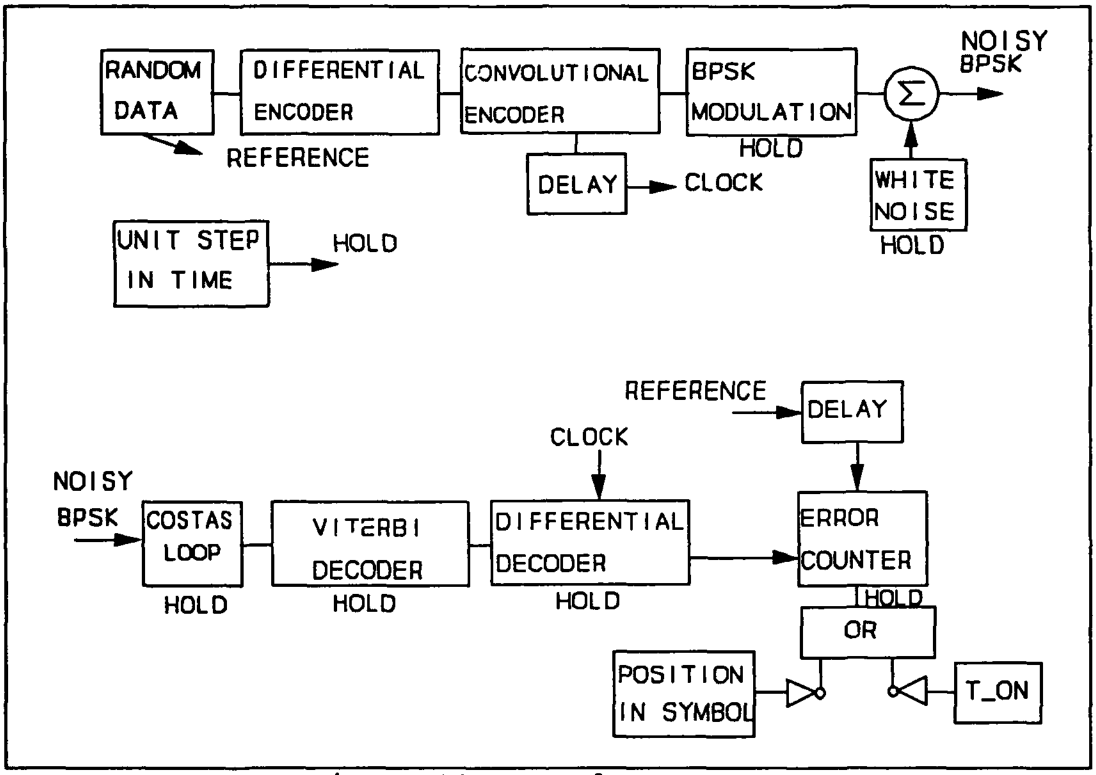

Figure 14. Complete System

signal sinks is shown in Figure 38 in Appendix A. The signal sinks stored values that were displayed with the Signal Display Editor (SDE) part of SPW™. A constant one replaced the noise input allowing verification of timing without noise. This allowed system delays and timing to be adjusted and verified. For P_b measurements, the signal sinks were removed and the only system output was a file containing data from the bit error counter block. The following sections describe the system signal flow, parameter configuration, timing considerations, testing process, and test results.

6.2.1 Signal Flow. The random data block produces random bits 0 for logic low and 1 for logic high at the system bit rate. The differential encoder block differentially encodes the random data, and the output goes to the convolutional encoder block. The convolutional encoder generates a rate 1/2 constraint length 7 encoded bit stream. The encoded bit rate is twice the uncoded bit rate, and the samples per encoded bit are half the number of samples per uncoded bit. The BPSK modulation block produces a baseband complex signal because the carrier parameter was set to 0 Hz internal to the block. The 0 to 1 input bit stream is changed to an antipodal ±1 bit stream internally in the BPSK modulation block. The modulated signal goes to a complex adder and the white noise given signal-to-noise ratio (SNR) and bandwidth block. The output of the noise block is the second input to the adder. The adder output is the noisy BPSK signal input to the Costas loop demodulator block. The quiescent frequency of the voltage controlled oscillator (VCO) was internally set to 0 Hz to match the 0 Hz carrier. The Costas loop real_arm or in-phase signal is the demodulated noisy signal. The Viterbi decoder block quantizes the noisy input signal and soft decision decodes the convolution encoded data. The encoded bit rate is twice the decoded bit rate, and the number of samples per encoded bit is half the number of samples per uncoded bit for the rate 1/2 code. The differential decoder block differentially decodes the Viterbi decoder block output. The differential decoder block output is then input to the bit error rate counter block. The second input to the bit error rate counter block is the original bit stream after a delay equal to the system delay.

6.2.2 System Parameters. The parameter Sfreq is the sampling frequency and is passed to all blocks except the delay blocks, the inverter block, and the differential encode and decode blocks. The SNR parameter is passed to the white noise generator block. The encoded rate parameter is passed to the white noise block and the Viterbi decoder block. The parameter bit rate is sent to the random data block and the convolutional encoder block. The loop frequency, VCO gain, and arm filter bandwidth are all passed to the Costas loop. The delay parameter is sent to both the differential encoder and decoder blocks. The block and initial error parameters are passed to the bit error rate counter. The T_on parameter is sent to both the T_on and position in symbol blocks. Table 13 summarizes the parameters and the parameter values.

6.2.3 Timing Considerations. The use of the convolutional encoder clock output for the Viterbi decoder clock input eliminates symbol synchronization concerns from the model. The remaining timing considerations deal with timing the simulation blocks for desired operation. The convolutional encoder produces its first valid output after a delay determined by Equation 51. The unit step block [1(t-T)] was used to hold the operation of the BPSK modulation, white noise, complex adder, and Costas loop

Table 13. System Parameters For 5 Samples/Encoded Bit Test

Parameter	Name	Value
sample frequency	Sfreq	24000 Hz
bit rate uncoded	bitrate	2400 bps
bit rate coded	encoded_rate	4800 bps
signal-to-noise ratio	SNR	3.5 dB
loop frequency	loopfreq	6.5087382 Hz
VCO gain	VCO_gain	130.17476 Hz/V
arm filter bandwidth	arm_band	9600 Hz
number of samples/uncoded bit	delay	-10
number of bits to count	block	100,000
first error event	initial_error	1
time to start error count	T_on	0.0205 sec

blocks until after valid data was available from the encoder. Another timing consideration was where in the bit period the Viterbi decoder should make a quantization sample. The approximate middle of the bit was chosen to allow for the Butterworth arm filters to settle. A bulk delay block was used to delay the clock output from the convolutional encoder by the necessary amount to produce a clock input signal that caused the Viterbi decoder to sample the middle of the input bits.

The final timing issues are related to the bit error counting block. For the comparison of the original data bits to the decoded received data bits to occur with proper timing, the original data bits must be delayed by an amount equal to the total system delay. The Viterbi decoder block introduces the majority of the system delay. This delay is determined by Equation 52. Additional delay is produced by the convolutional encoder as previously discussed. A bulk delay block is used to delay the original data by the system delay. The data bits input to the error counter are either 0 for a logic low or 1 for a logic high and are ideal signals. The sample point for comparing the bits is arbitrary. The position in symbol block by default uses the midpoint of the bit based on the bit rate and sample frequency. The position in symbol block is used to allow the bit error counter to compare at the midpoint of the bit and held the block operation between midpoints. The T_ON block is used to hold the error counter block operation until after the total delay time. For the system simulation with the parameter values given in Table 13, Equations 51 and 52 predict delays of 4 samples and 485 samples respectively. An additional 2 sample delay is caused by the delay of the convolutional encoder clock output before its use as the clock input for the Viterbi decoder. The total system delay is 491 samples measured from the first sample of the first bit output from the random data block to the first sample of the first bit output from the differential decoder block. The value for the T_ON parameter is found by dividing the total system delay plus one more sample by the sample frequency. The one additional sample starts the error counter block after the first data bit transition. The value for the T_ON parameter is 492/24000 = 0.0205 sec. As recommended in [10], a test run of the system was done with a reasonable number of samples (5000) for this test. Additional signal sink blocks were used to store all timing signals, and the Signal Display Editor was used to verify correct timing in relation to the stored data bits. The full system is shown in Figure 39 in Appendix A.

6.3 Testing Process

The performance measure for the system is probability of bit error P_b given a known energy per bit E_b to noise N_o ratio. The white noise block was used to generate additive white Gaussian noise that was zero mean with a variance adjusted to produce the specified signal-to-noise ratio. At the point in the system where the white noise block is located, encoded bits at the encoded rate are present. The white noise block is operated with the assumption that the noise bandwidth equals the bit rate of the data entering the block. As discussed in [4], the energy per bit before the rate 1/2 convolutional encoder is 3 dB more than the energy per bit after the encoder. However, the Costas loop arm filters were set to twice the received bit rate as recommended in [15]. This makes the noise bandwidth twice what the white noise block operation assumes. The net effect is that the 3 dB more E_b is canceled by the 3 dB more noise bandwidth. This made setting the SNR parameter for the white noise block to 3.5 dB equivalent to having a 3.5 dB E_b-to-N_o ratio into the Costas loop demodulator. With the SNR set, the only remaining parameter to set before the system is ready for simulation is the block length. The block length is determined by how many bits are to be counted between error counter writes to the results file. By setting the block length parameter to the total number of bits, for example 10^5, one error count value and one bit count value are stored after each group of 10^5 bits are compared. For testing, the block length was set equal to 10 divided by the order of magnitude of the expected error rate. The expected error rates for the system were shown and discussed in Section 3.2. The requirement for processing $10/P_b$ bits was discussed in Section 3.3.1 No other outputs were required from the system, so all signal sinks were removed from the system.

A sample instant is considered an iteration when the simulation is run. For data bits with 10 samples per bit, 10 iterations must be processed for each bit. Therefore, to simulate 10^5 bits, the simulation must run for 10^6 iterations if no delays are present in the simulation. The modem model system delay was 491 samples. Each SNR ratio was simulated for a number of iterations equal to 10 times the block length plus 500 iterations. Table 14 shows the four levels of SNR, block length, and number of iterations used for testing.

Table 14. Number of Iterations For 5 Samples/Encoded Bit Test

SNR	Block Length	Number of Iterations
3.5 dB	10^5	1.0005×10^6
4.5 dB	10^6	1.00005×10^7
5.0 dB	10^6	1.00005×10^7
5.5 dB	10^7	1.000005×10^8

BPSK matched filter testing, Section 4.5, showed that over one and a half hours were required to simulate 10^5 bits with $1.0001 \text{X} 10^6$ iterations in the interactive mode of the Block Diagram Editor (BDE). Therefore, another approach was taken for the modem model testing. The BDE tool for code generation produces a file containing a C language program. The compiled C code can be run on computers other than the system with the SPW™ software, provided the required SPW™ library files are available on the other computer [10]. As a test for consistency of results, the BPSK matched filter system code was generated and run. The time for the simulation was reduced from over an hour and a half to approximately 25 minutes with the same resulting error count. At the time the C code program is started, the number of iterations and noise seed value can be entered on the command line. With a UNIX script file, the ten noise seeds for each value of iteration count can be looped through.

6.3.1 Initial Test Results. The simulation of the modem model was first done with system parameters as shown in Table 13 and Table 14. The number of iterations per uncoded data bit was ten, whereas after the convolutional encoder, there were five iterations per encoded bit. The results of the simulations are shown in Table 27 through Table 30 in Appendix B. Table 15 shows the data statistics.

Table 15. $\hat{P}_b$ Statistics For 5 Samples/Encoded Bit

E_b/N_o	μ	σ^2	σ
3.5 dB	3.960×10^{-3}	6.95111×10^{-8}	2.63650×10^{-4}
4.5 dB	5.314×10^{-4}	3.24627×10^{-9}	5.69760×10^{-5}
5.0 dB	1.702×10^{-4}	5.97733×10^{-10}	2.44486×10^{-5}
5.5 dB	5.542×10^{-5}	2.89107×10^{-11}	5.37686×10^{-6}

6.3.2 Five Samples/Encoded Bit Result. The mean value of the ten simulation runs was plotted for each E_b/N_o level. Additionally, vertical lines representing the value of $\pm\sigma$ at each E_b/N_o level were plotted. In Figure 15, the resulting curve is labeled measured data (5). Also shown, for comparison, are the

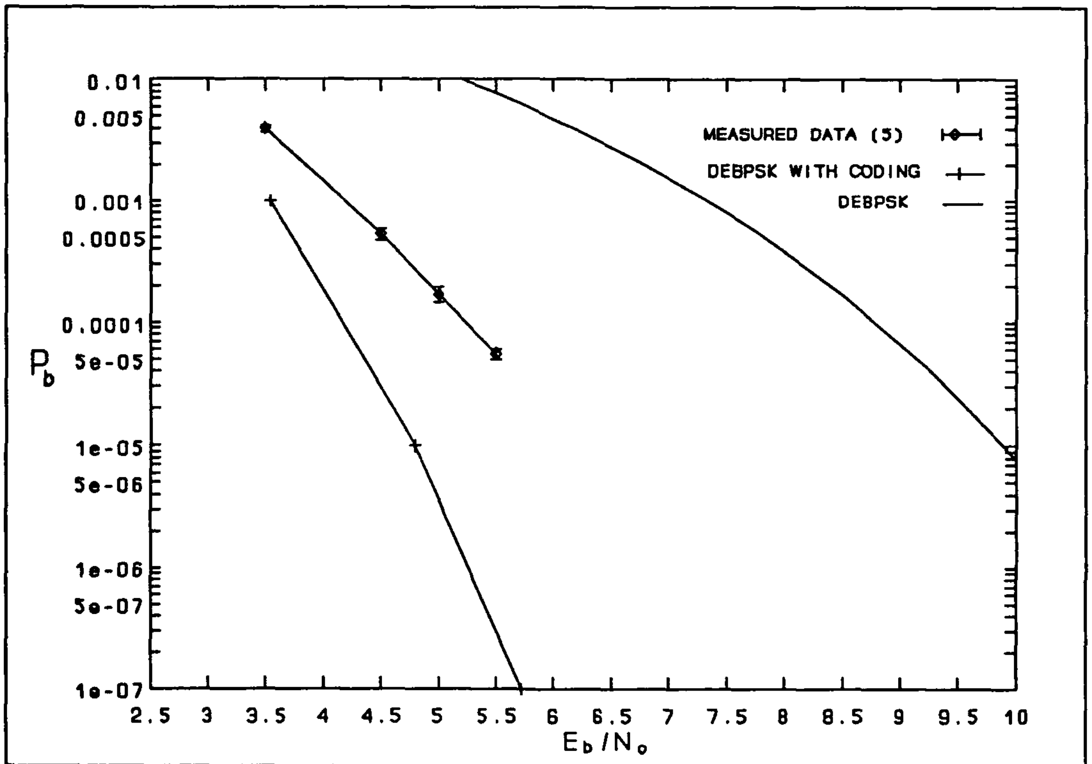

Figure 15. Theoretical and Measured Error Rate Plots for 5 Samples per Encoded Bit

theoretical P_b curves for DEBPSK as discussed in Section 3.1 and DEBPSK with coding discussed in Section 3.2. Figure 15 indicates that the simulation required approximately 1 dB higher E_b/N_o than the theoretical line to place P_b in the 10^{-4} range. The code gain for the simulation with five samples/encoded bit is shown in Table 16. The gain is measured as the difference between the

E_b/N_o required to produce a given P_b from Equation 9 (DEBPSK theory) and the E_b/N_o of the simulation that produced the given P_b.

Table 16. Code Gain For 5 Samples/Encoded Bit Test

E_b/N_o	P_b	Code Gain
3.5 dB	3.960×10^{-3}	2.6797 dB
4.5 dB	5.314×10^{-4}	3.2822 dB
5.0 dB	1.702×10^{-4}	3.4926 dB
5.5 dB	5.542×10^{-5}	3.5993 dB

6.3.3 Additional Testing. The first simulations of the full system were run with ten samples per uncoded bit, resulting in five samples per coded bit. The effects of encoding and whether the number of samples per bit in all simulation blocks should be maintained at the 8 to 16 level were not discussed in [2]. In order to determine if having five samples per encoded bit was affecting the system P_b performance, the system would require modification to achieve 10 samples per encoded bit. The simulation could be modified in two ways to achieve 10 samples per encoded bit. The following sections discuss both methods of modification and the results of simulation.

6.3.3.1 Increased Sample Frequency. Increasing the sample frequency in the simulation from 24000 samples/sec to 48000 samples/sec produces the desired 10 samples per encoded bit. The system timing was adjusted to account for the change in sample frequency. The clock output from the convolutional encoder must now be delayed by 5 samples to cause the Viterbi decoder to operate on the midpoint of the encoded bit. The unit step block was set to transition on the 9th sample as predicted by Equation 51. The total system delay was calculated as in Section 6.2.3 to be 984 samples. The parameter T_ON remains 0.0205 sec. Finally, the delay parameter passed to the differential encoder and decoder was changed to 20 because there were 20 samples per uncoded bit. Figure 40 in Appendix A shows the system. The C code for the modified system was generated and the simulation run for a SNR of 3.5 dB. Table 32 in Appendix B shows the results. Table 17 shows the statistics of the results.

Table 17. $\hat{P}_b$ Statistics For 48000 Hz Sample Frequency: 3.5 dB E_b/N_o

μ (10^{-3})	σ^2 (10^{-8})	σ (10^{-4})
1.924	4.62933	2.15159

Although simulation time was not a area of study for this thesis, it became a concern. With 20 samples per uncoded bit, the number of iterations for the SNR of 3.5 dB simulation was 2.001×10^6. Over one and a half hours were required for each noise seed simulation if the process had sole use of the computer.

6.3.3.2 Decrease Bit Rate. Instead of raising the sample frequency, the data rate can be lowered to reach 10 samples per encoded bit. The bit rate parameter change from 2400 bps to 1200 bps caused the need to change the arm_band parameter from 9600 Hz to 4800 Hz. System delays were changed to the same values that resulted from doubling the sample frequency as discussed previously. The T_ON parameter was changed to 0.041 sec to account for the fact that at 1200 bps it took twice as long to produce the same number of bits as were produced at 2400 bps. The modified system is shown in Figure 41 in Appendix A.

The C code for the modified system was generated and the simulation run for a SNR of 3.5 dB. Again, $2.001X10^6$ iterations were processed to produce $1X10^5$ data bits. Table 33 in Appendix B shows the results. Table 18 shows the statistics of the results.

Table 18. $\hat{P}_b$ Statistics For 1200 bps Data Rate: 3.5 dB E_b/N_o

μ (10^{-3})	σ^2 (10^{-8})	σ (10^{-4})
1.842	8.33289	2.88667

<u>6.3.3.3 Ten Samples/Encoded Bit Test.</u> The two methods of simulation with 10 samples per encoded bit through the Costas loop produced nearly the same probability of bit error for a E_b/N_o of 3.5 dB. The difference between the probability of bit error result for each method was less than the value of σ for either method. Based on this fact, the remaining simulations were run with the system that achieved 20 samples per uncoded bit by lowering the data rate. An added consequence of using this method is that the Costas loop is configured as it was for testing described in Section 5.5.

The simulation was run for each of the 10 noise seeds at the four SNR levels. Table 19 shows the four SNR levels, block length, and number of iterations. Table 20 shows the system parameters.

Table 19. Number of Iterations for 10 Samples/Encoded Bit Test

SNR	Block Length	Number of Iterations
3.5 dB	10^5	2.001×10^6
4.5 dB	10^6	2.0001×10^7
5.0 dB	10^6	2.0001×10^7
5.5 dB	10^7	2.00001×10^8

Table 20. System Parameters For 10 Samples/Encoded Bit Test

Parameter	Name	Value
sample frequency	Sfreq	24000 Hz
bit rate uncoded	bitrate	1200 bps
bit rate coded	encoded_rate	2400 bps
signal-to-noise ratio	SNR	3.5 dB
loop frequency	loopfreq	6.5087382 Hz
VCO gain	VCO_gain	130.17476 Hz/V
arm filter bandwidth	arm_band	4800 Hz
number of samples/uncoded bit	delay	-20
number of bits to count	block	100,000
first error event	initial_error	1
time to start error count	T_on	0.041 sec

The simulation time using compiled C code ranged from approximately 1.5 hours for $2.001X10^6$ iterations to about 27 hours for $2.0001X10^8$ iterations. The times given above are for a single noise seed and a SPARC-2™ computer. With three computers running simulations with different noise seeds, over four days of computer time were required to simulate the system at the 5.5 dB SNR level for all 10 noise seeds. The simulation results are shown in Table 35 through Table 38 in Appendix B. Table 21 shows the statistics of the results.

Table 21. $\hat{P}_b$ Statistics For 10 Samples/Encoded Bit

E_b/N_o	μ	σ^2	σ
3.5 dB	1.842×10^{-3}	8.33289×10^{-8}	2.88667×10^{-4}
4.5 dB	2.332×10^{-4}	6.75733×10^{-10}	2.59949×10^{-5}
5.0 dB	7.220×10^{-5}	2.47511×10^{-10}	1.57325×10^{-5}
5.5 dB	1.986×10^{-5}	4.49822×10^{-12}	2.12090×10^{-6}

6.3.4 Ten Samples/Encoded Bit Result. The mean value of the ten simulation runs was plotted for each E_b/N_o level. Vertical lines representing the value of $\pm\sigma$ at each E_b/N_o level were also plotted. In Figure 16, the resulting curve is labeled measured data (10). The other curves are the same as shown in Figure 15. The simulation with 10 samples per encoded bit requires approximately 0.7 dB higher E_b/N_o than the theoretical line to place P_b in the 10^{-4} range. Table 22 gives the code gain for this simulation.

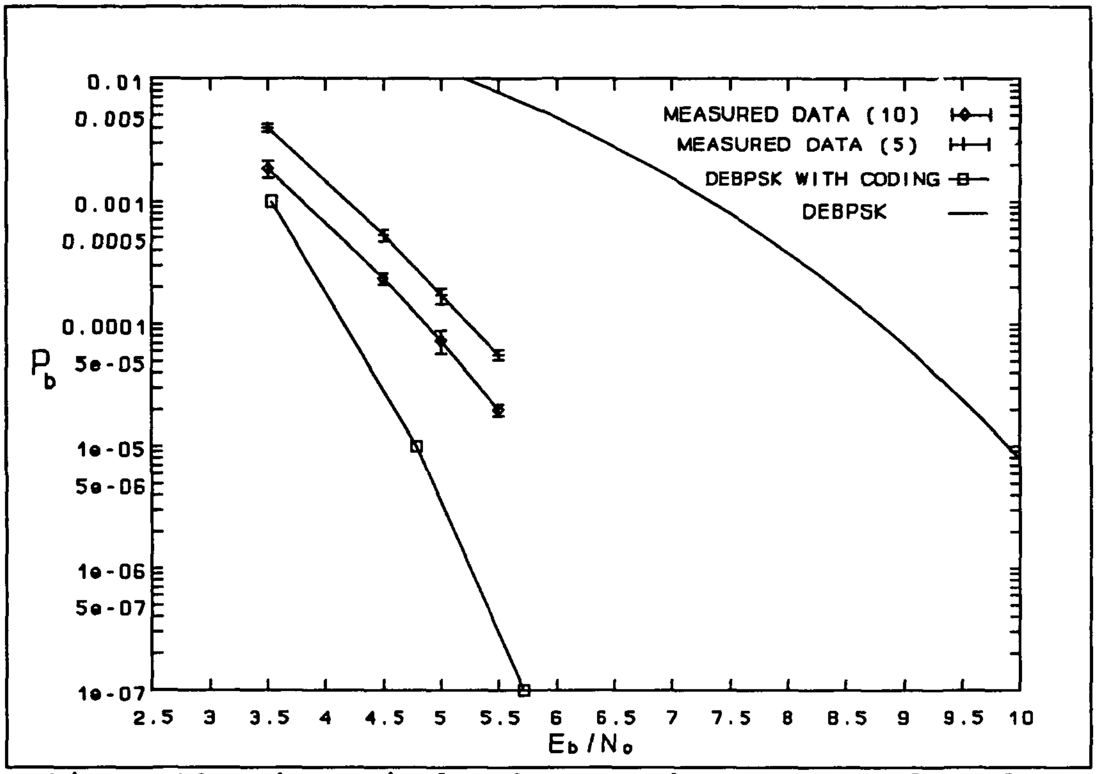

Figure 16. Theoretical and Measured Error Rate Plots for 5 and 10 Samples per Encoded Bit

Table 22. Code Gain For 10 Samples/Encoded Bit Test

E_b/N_o	P_b	Code Gain
3.5 dB	1.842×10^{-3}	3.3570 dB
4.5 dB	2.332×10^{-4}	3.8068 dB
5.0 dB	7.220×10^{-5}	3.9633 dB
5.5 dB	1.986×10^{-5}	4.0914 dB

6.4 Results Discussion

The improvement in the simulation performance can be seen in Figure 16. The 10 samples per encoded bit P_b is approximately

0.4 dB closer to the theoretical line than the result of testing with 5 samples per encoded bit. The theoretical signal-to-noise ratio for the loop (ρ_1) ranged from approximately 19.5 dB to 21.5 dB as discussed in Chapter V. As shown by [5], and discussed in Chapter III, the P_b vs E_b/N_o curve for this ρ_1 should be steeper and differ from the theoretical case of $\rho_1 = \infty$ by less than 0.5 dB. The P_b curve resulting from the simulation appears to agree with a ρ_1 on the order of 13.5 dB. The actual ρ_1 of the loop simulation was not found by the testing done in this thesis. Simulations that would allow determination of the actual Costas loop ρ_1 must be performed before an exact determination of the effect of ρ_1 on the results would be possible.

6.5 Summary

This chapter first covered the configuration and testing of the coding blocks. The ability of the coding blocks to encode and decode a bit stream without error in a noiseless test verified the correct configuration of these blocks. The full model was then described with a detailed discussion of timing considerations. The majority of the system delay, over 98%, was the result of the convolutional encoder and Viterbi decoder blocks. The initial test result was improved by approximately 0.4 dB by doubling the number of samples per encoded bit processed by the Costas loop. Both increasing the sample frequency or decreasing the data rate doubled the number of samples per encoded bit and resulted in an improvement in E_b/N_o required to achieve a given P_b. The full modem simulation resulted in a code gain of between 3.357 dB and 4.0914 dB for E_b/N_o values from 3.5 dB to 5.5 dB. These results differ from the theoretical code gain by approximately 0.7 dB for P_b in the 10^{-4} range. It is unknown if ρ_1 contributed to the difference from theoretical, because testing performed did not calculate the actual ρ_1 of the simulated Costas loop. Before these results are used as more than a rough estimate of the CQM-248 performance, further refinement of the simulation should be done. Additional testing of the Costas loop and refinements to the simulation are proposed in Section 7.3.

VII. Conclusions and Recommendations

7.1 Conclusions

The goal of this thesis was to estimate the CQM-248 modem performance by computer simulation. The model of the modem includes differential and convolutional encoding of the data with BPSK modulation of the encoded data. The channel was modeled as a AWGN baseband disturbance. A second-order Costas loop demodulates the received noisy data and soft decision Viterbi decoding is used. A differential decoder provides the final data output for comparison to the original data. Accomplishing this goal required three objectives.

The first objective was to gain experience with the SPW™ tool. The BPSK matched filter detection system was used to gain experience with the SPW™ tool. Also, testing the matched filter system demonstrated the Monte Carlo error estimation method. Most importantly, the matched filter system verified the operation of the bit error counter and white noise generation blocks.

The matched filter detection system proved useful as a learning tool. Simulation timing issues were discovered and solved. For example, the delay of the original data for comparison with the received data was not obvious. The system worked as expected with P_b estimated results within one standard deviation of theoretical.

The second objective was to develop and demonstrate proper performance of a Costas loop demodulator. The loop response to a phase step or a frequency step was simulated to verify loop operation in a noiseless case. The loop response was compared to theoretical predictions.

The Costas loop with the corrected equations in the loop filter was used in further simulations. The loop rise time to peak deviation in response to a phase step was within 2.9% of theoretical. The peak deviation in response to a phase step was within 7.7% of theoretical. For a step in frequency, the loop steady state phase error was within 5.5% of theoretical, and lock-in occurred within one cycle of the loop natural frequency. This comparison to theoretical predictions demonstrated loop performance in a noiseless test.

The final objective was to assemble and test a full system. The complete model was assembled after the Costas loop was correctly working. The full system included differential encoding and decoding; convolutional encoding with eight-level soft Viterbi decoding; BPSK modulation of the encoded data; an AWGN channel; and the Costas loop demodulator. Two ratios of samples per bit were used in the testing and four E_b/N_o levels were tested.

The full modem simulation resulting P_b was improved by approximately 0.4 dB by doubling the number of samples per encoded bit processed by the Costas loop from 5 samples per encoded bit to 10 samples per encoded bit. However, the P_b curve still differed from the theoretical curve by about 0.7 dB for P_b in the 10^{-4} range. As discussed in Chapter III and Chapter VI, the loop signal to noise ratio affects the slope of the P_b curve. Because, the actual loop signal-to-noise ratio was not determined for the simulation, the effect on the simulation result is unknown. The full modem simulation resulted in a code gain of between 3.36 dB and 4.09 dB for E_b/N_o values from 3.5 dB to 5.5 dB. Further refinement of the simulation should be done before these results are used as more than a rough estimate of the CQM-248 performance.

7.2 Recommendations

The following paragraphs are recommendations for follow-on work.

1. Testing of the Costas loop should be done in a noisy environment. The simulated Costas loop signal to noise ratio should be determined and compared to the theoretical predictions in Chapter V. This comparison could lead to further modifications to the Costas loop that would affect P_b of the full system simulation.

2. The model should be extended to a coded QPSK system. This would provide another measure of modem performance. The building block for the extension to QPSK would be the modification of the Costas loop to simulate a four-phase Costas loop. Phase and frequency step responses of the Costas loop should again be simulated and compared to theoretical predictions.

3. When the CQM-248 modem is available, hardware testing should be done. The P_b vs E_b/N_o results should then be compared to the simulation.

Appendix A

The plots included in this appendix were generated by the Block Diagram and Signal Display Editor. Figures 17 through 25 and Figures 34 through 41 were generated by the Block Diagram Editor tool from SPW™. Figures 26 through 33 were created by the Signal Display Editor tool from SPW™. Figure 17 is the BPSK matched filter system. Figures 18-20 are the detail block diagrams used in the matched filter system. Figures 21-25 are the Costas loop detail and the diagrams of the systems used to test the loop. Figures 26-33 are the Costas loop test system outputs. Figures 34 and 35 are the detail and symbol for the differential encoder. Figures 36 and 37 are the detail and symbol for the differential decoder. Figures 38-41 are block diagrams of the systems used for full system testing.

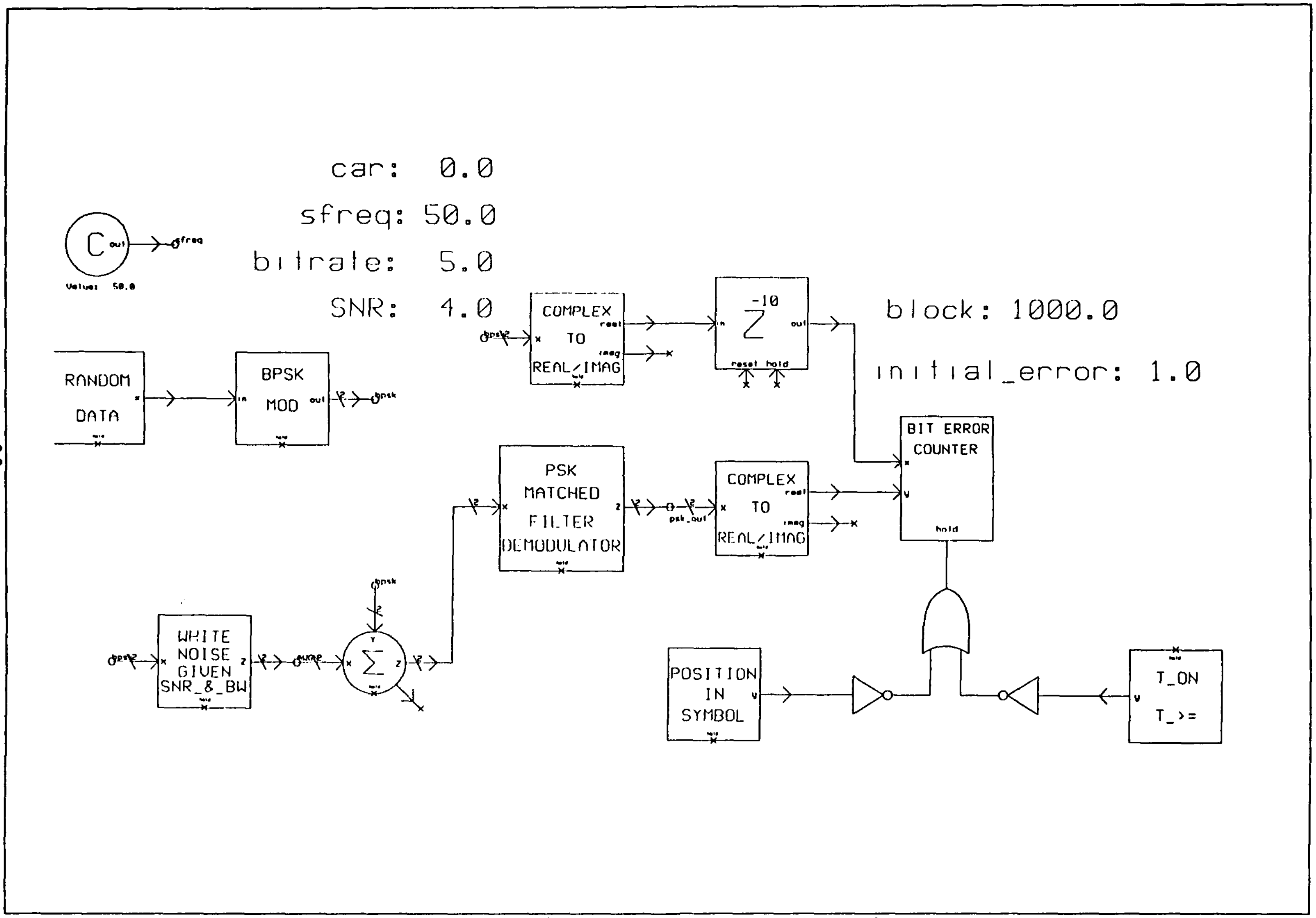

Figure 17. Block Diagram of BPSK Matched Filter System

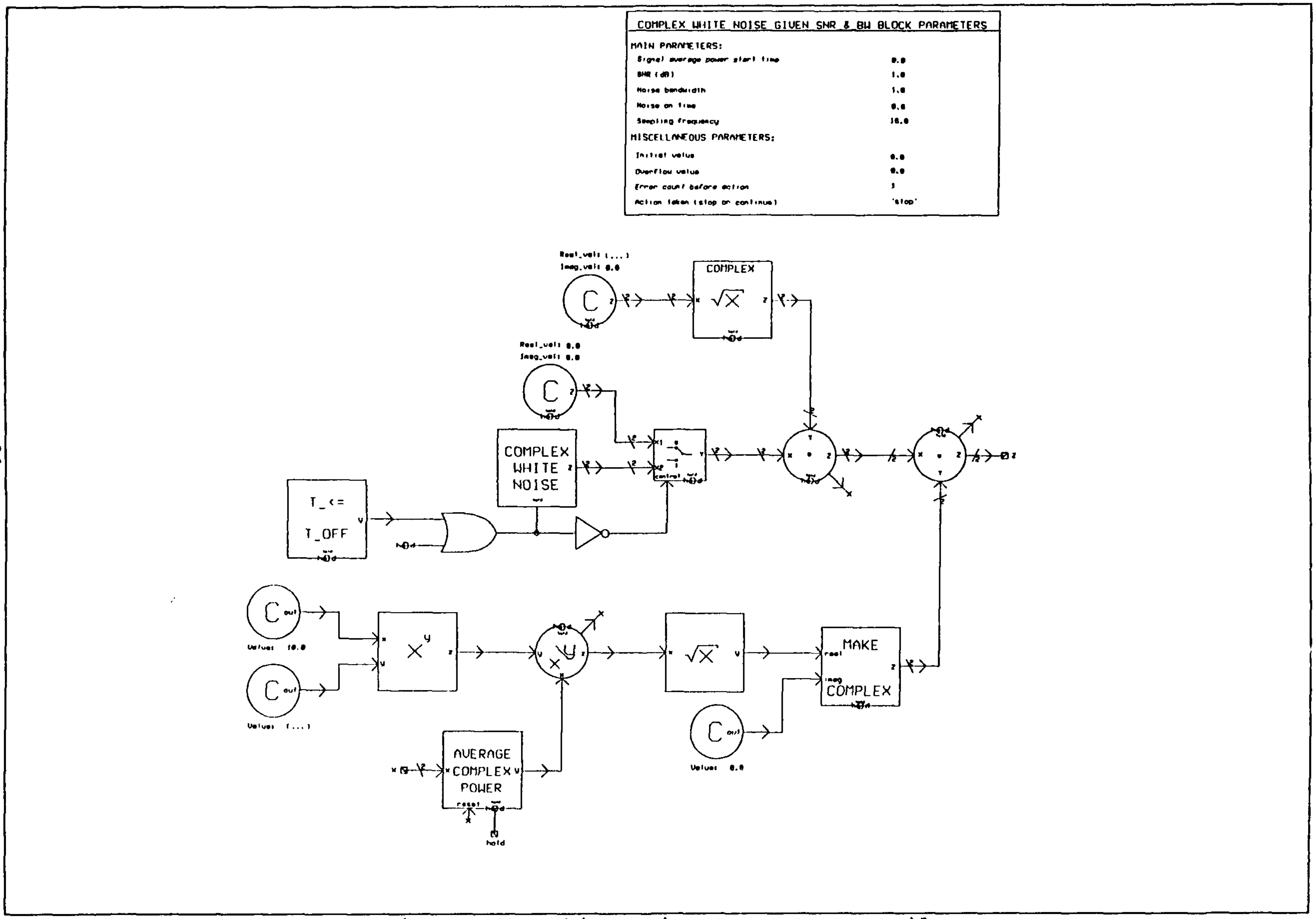

Figure 18. White Noise Generator Detail

PSK MATCHED_FILTER DEMODULATOR BLOCK PARAMETERS	
MAIN PARAMETERS:	
PSK constellation first angle (Deg.)	45.0
PSK modulation order	4.0
Baud rate	1.0
Time delay to input (Sec.)	0.0
Channel phase rotation (Deg.)	0.0
Sampling frequency	16.0
MISCELLANEOUS PARAMETERS:	
Initial value	0.0
Overflow value	0.0
Error count before action	1
Action taken (stop or continue)	'stop'

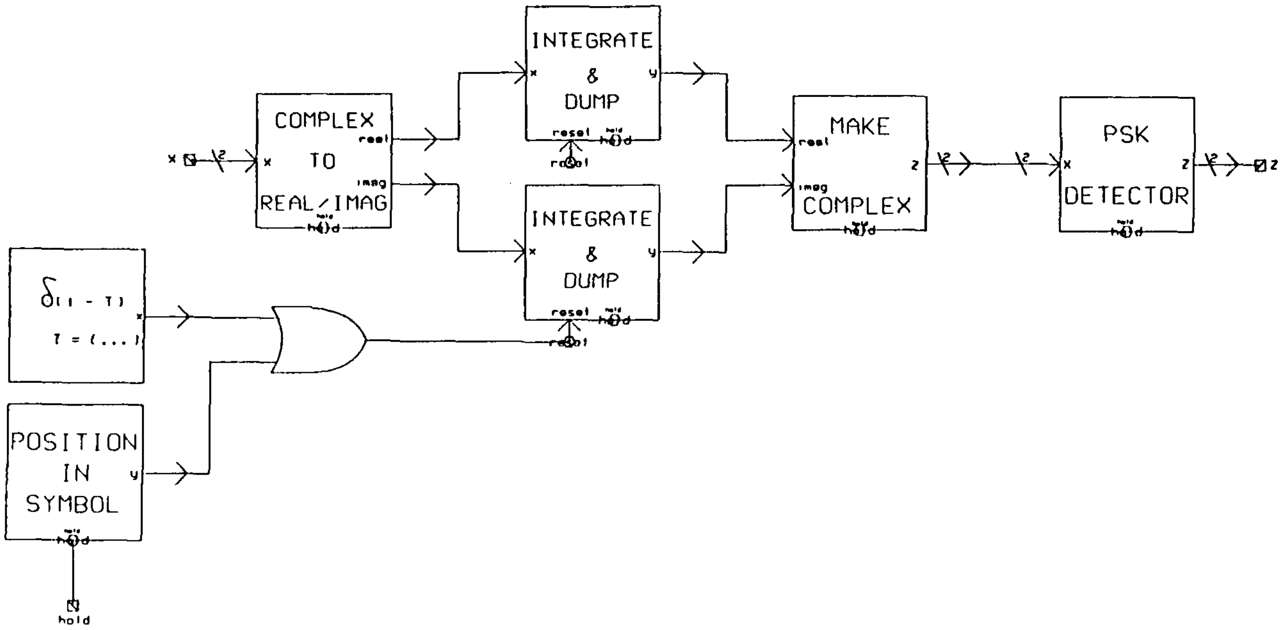

Figure 19. PSK Matched Filter Demodulator Detail

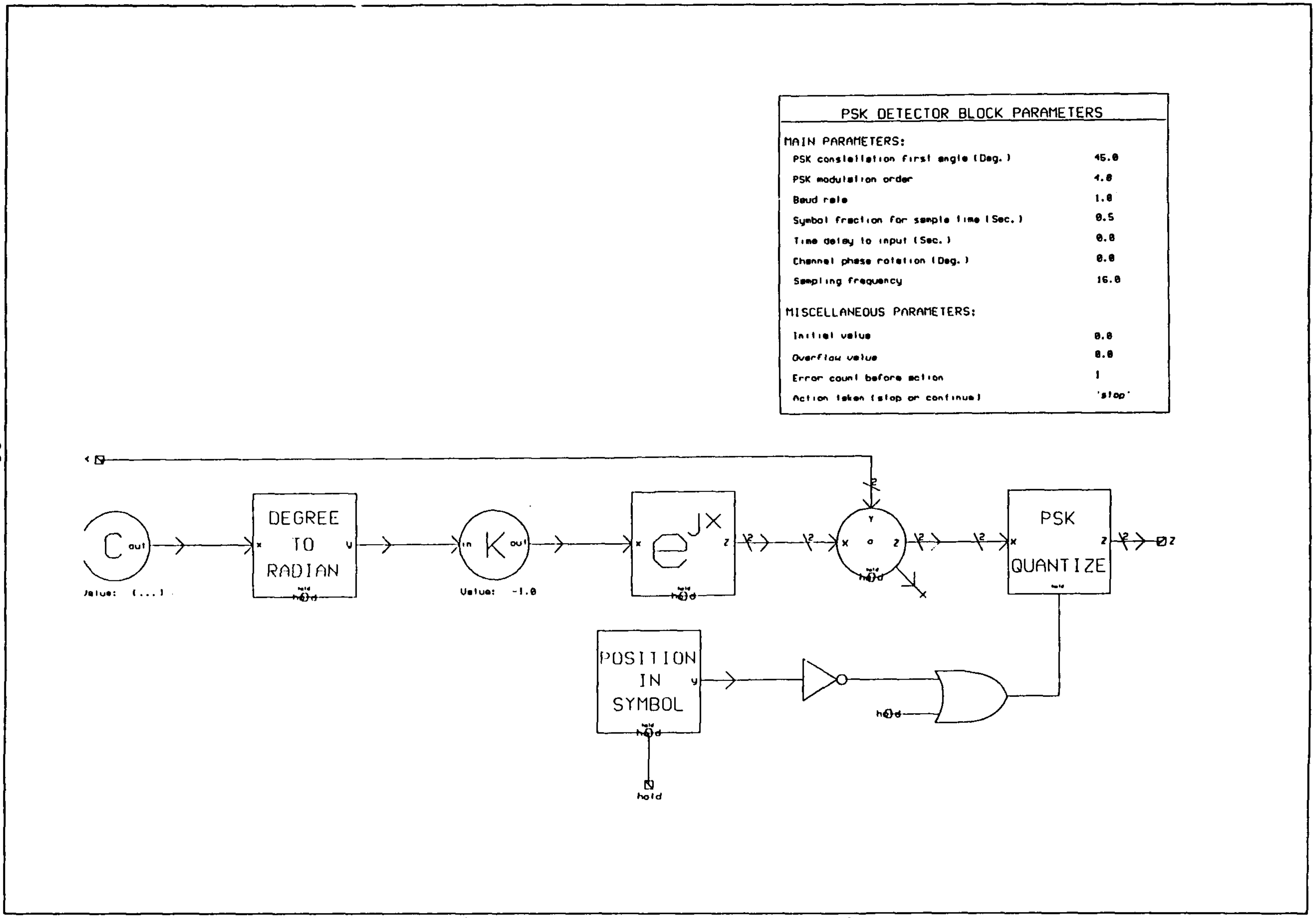

Figure 20. PSK Detector Detail

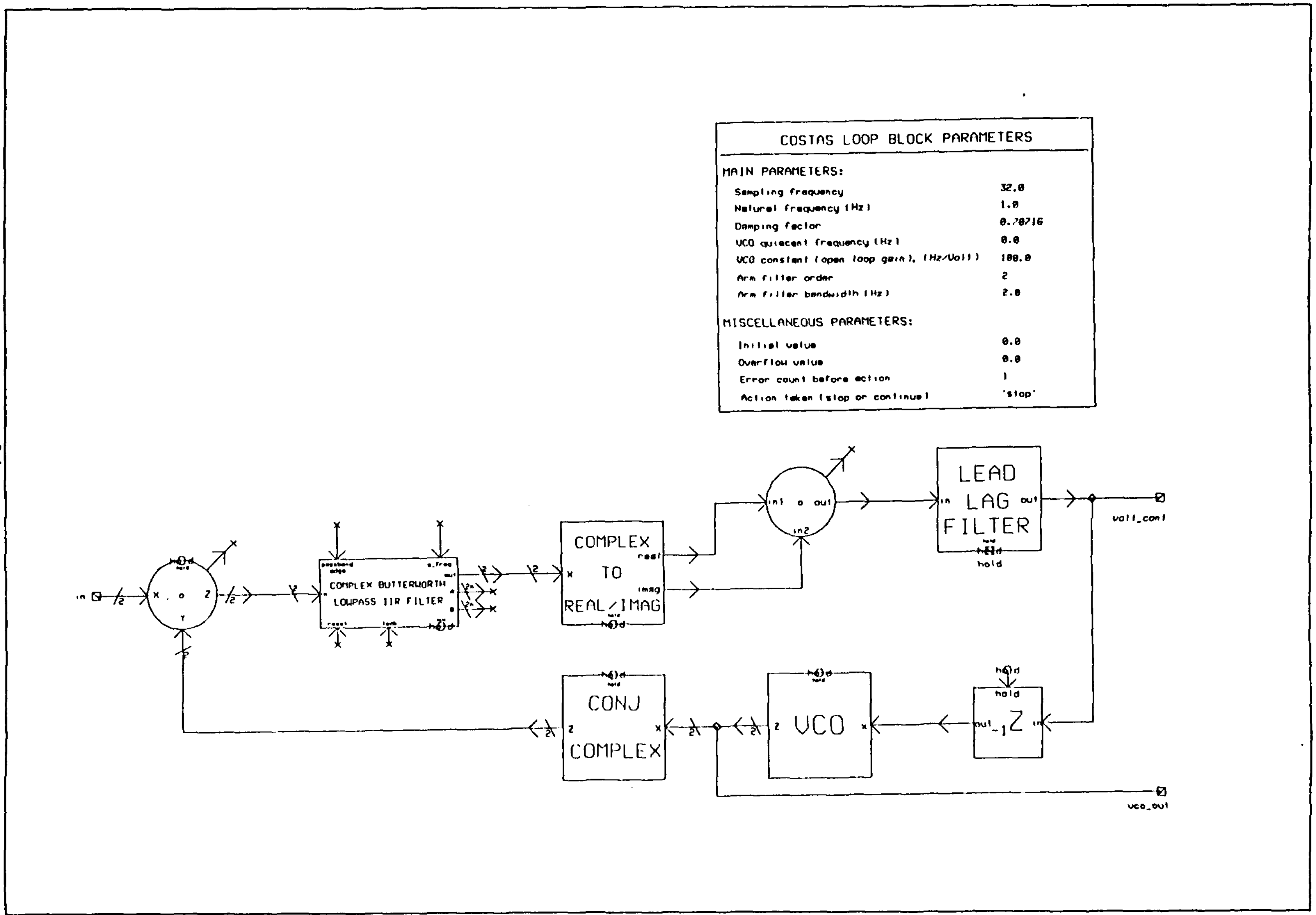

Figure 21. Costas Loop Detail

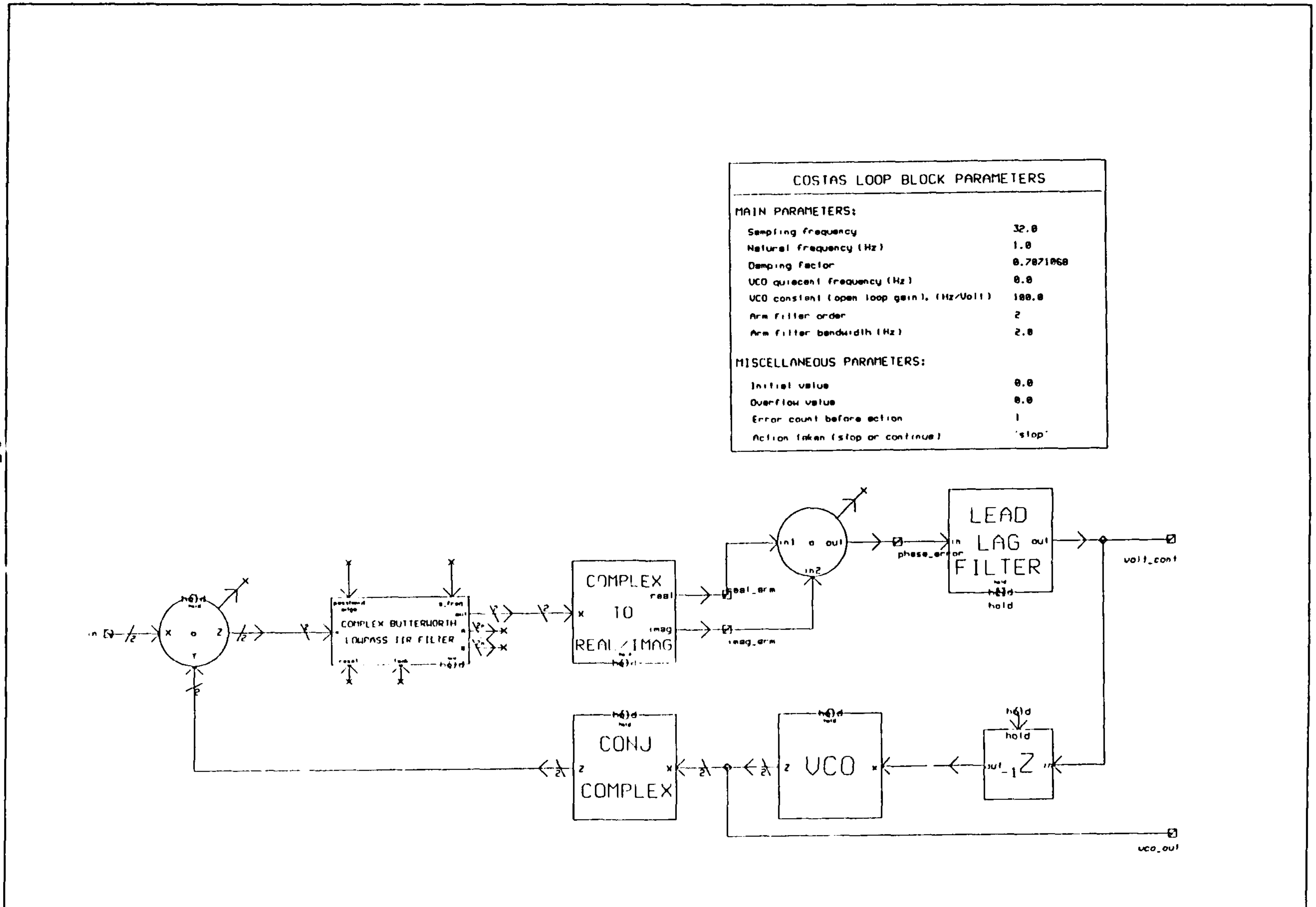

Figure 22. Modified Costas Loop Detail

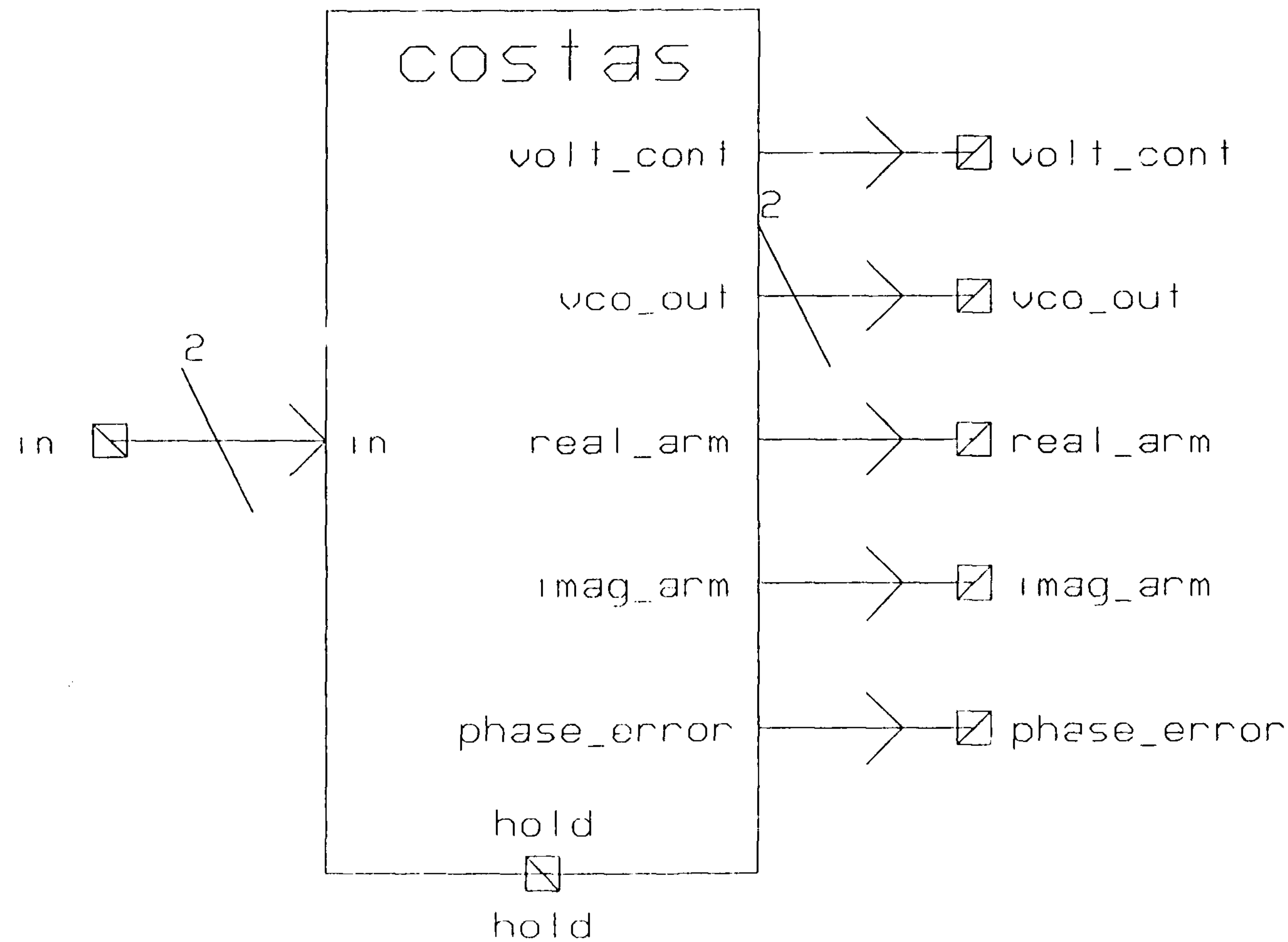

Figure 23. Costas Loop Symbol

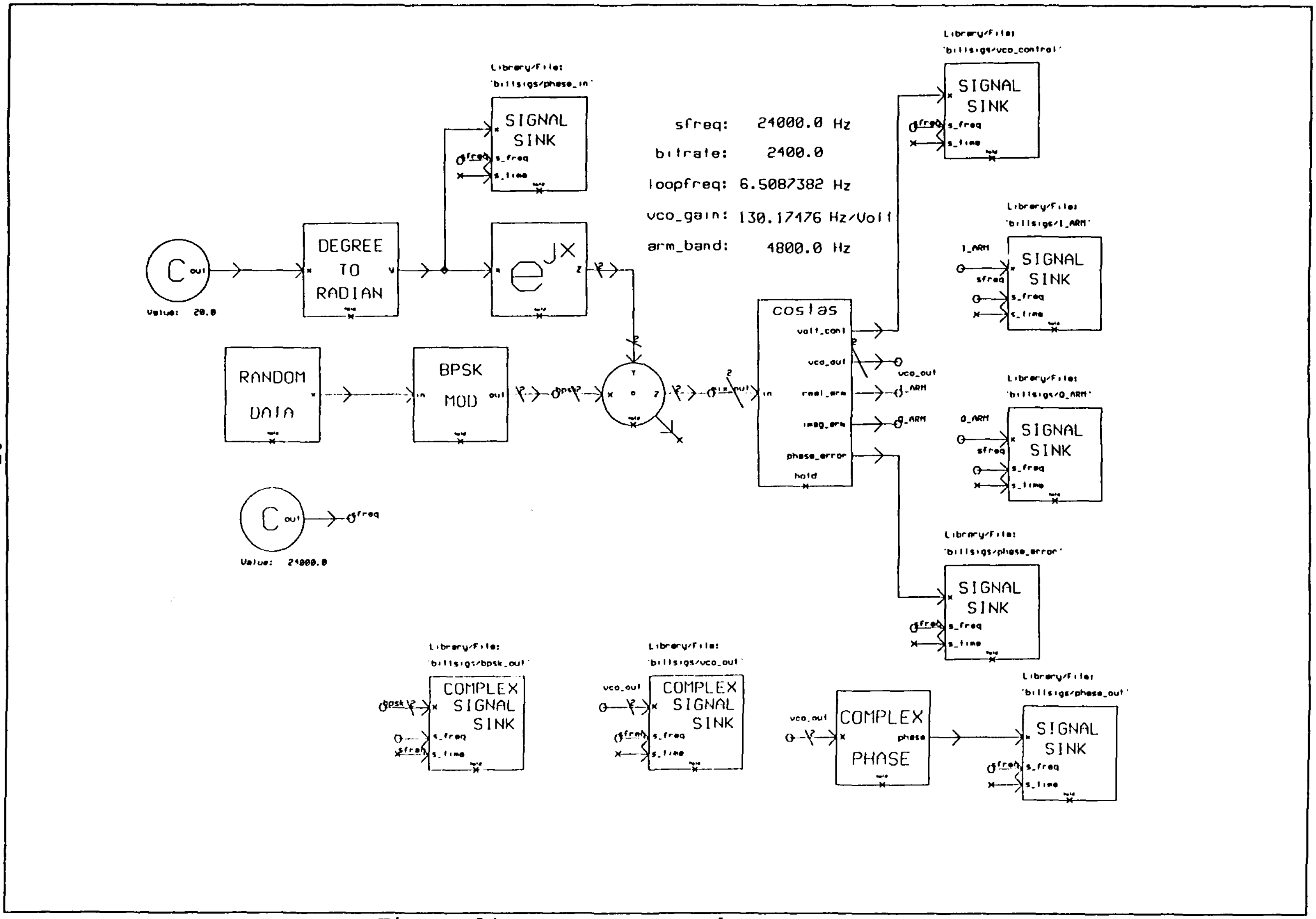

Figure 24. Costas Loop Phase Step System

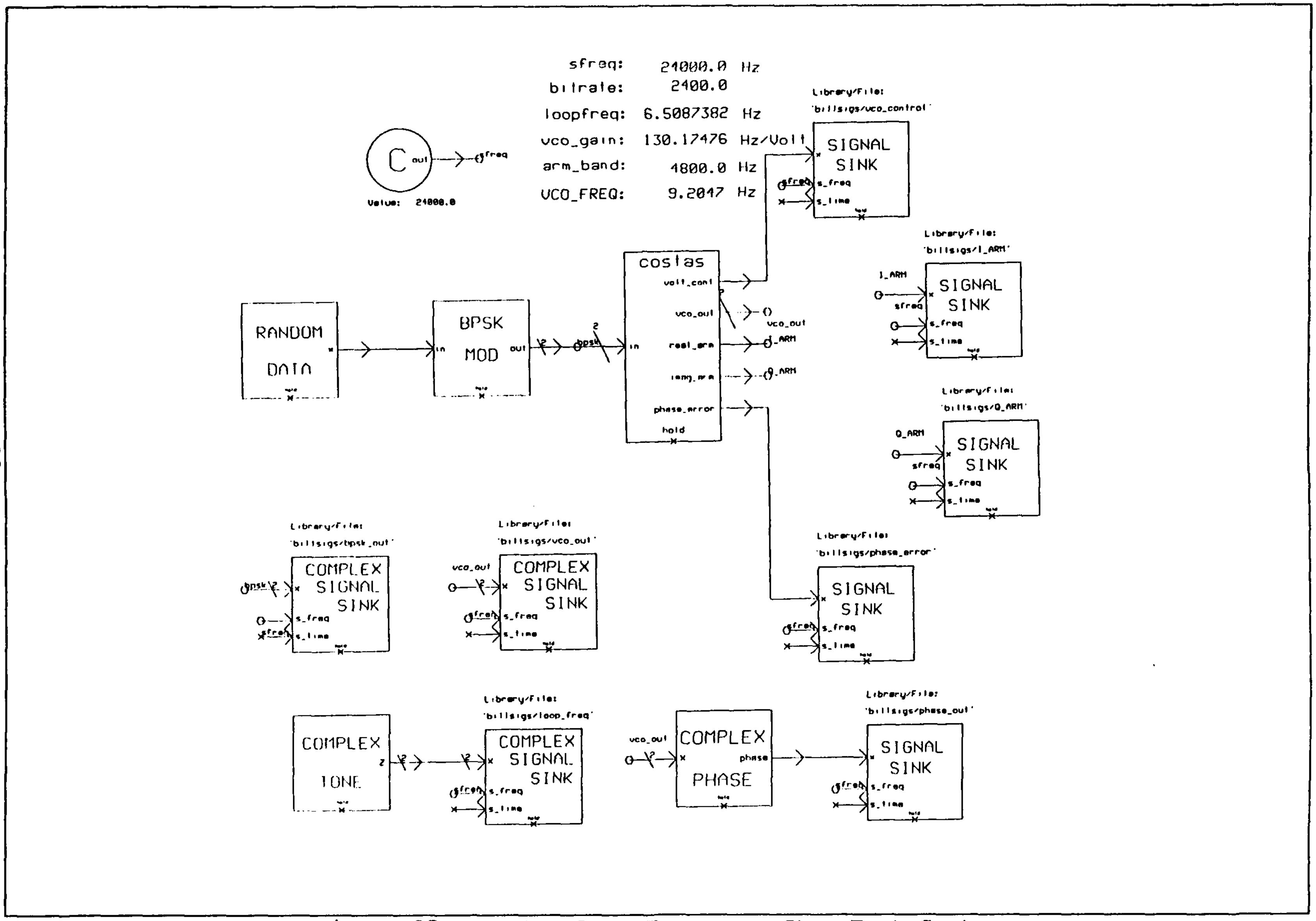

Figure 25. Costas Loop Frequency Step Test System

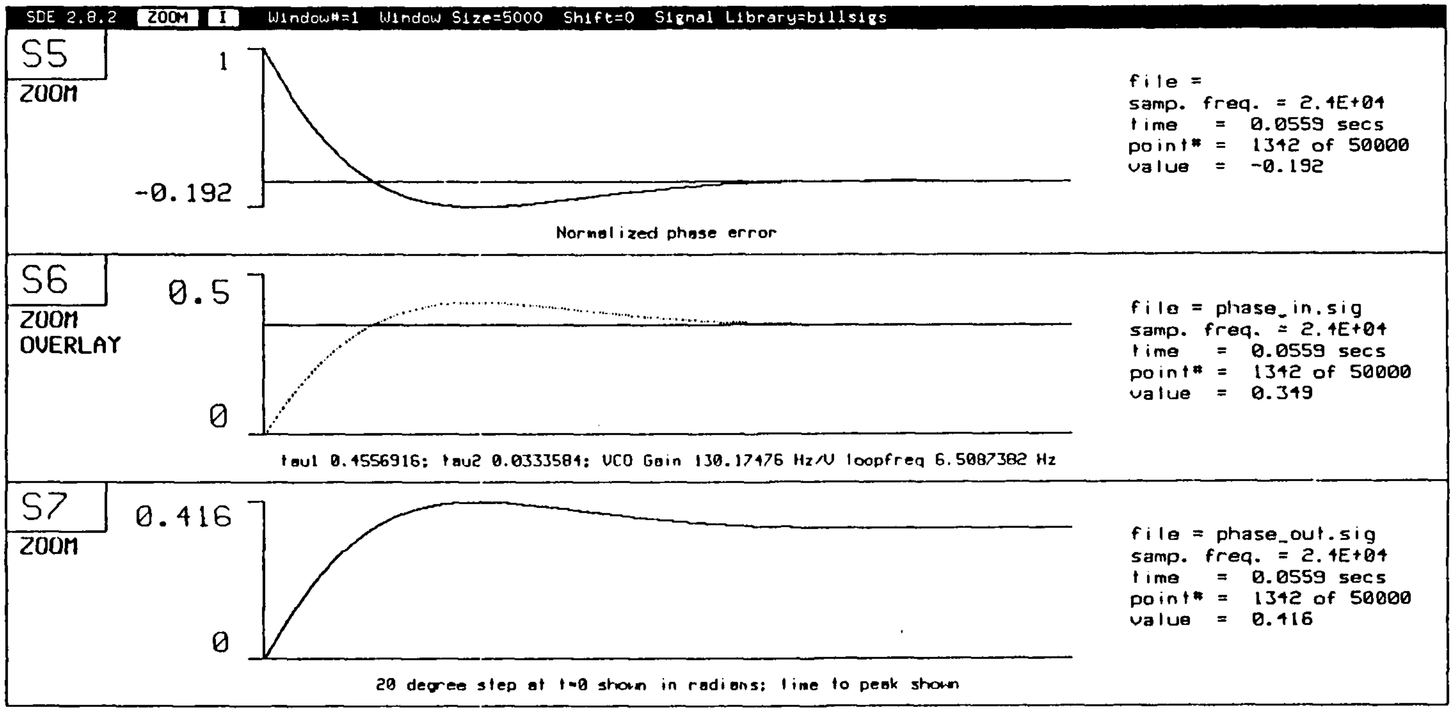

Figure 26. Costas Loop Response to Phase Step Plot 1

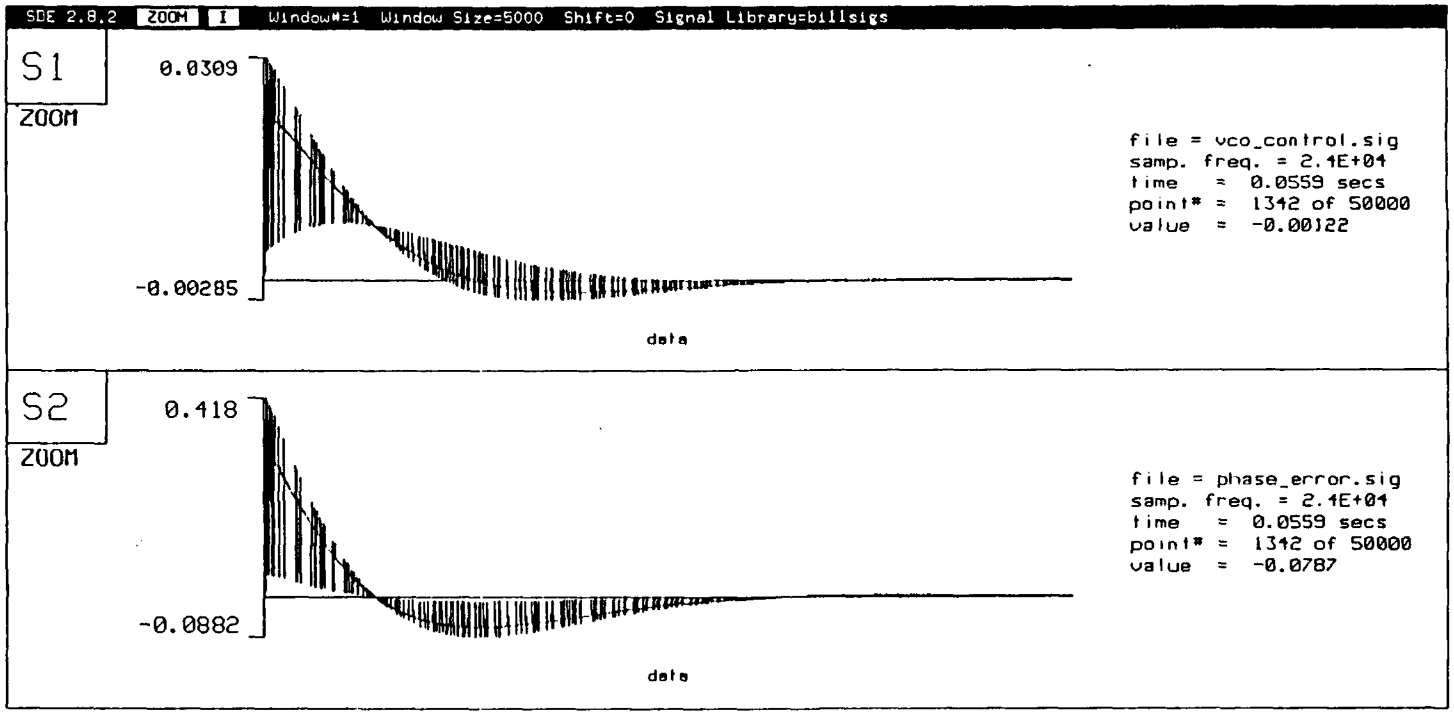

Figure 27. Costas Loop Response to Phase Step Plot 2

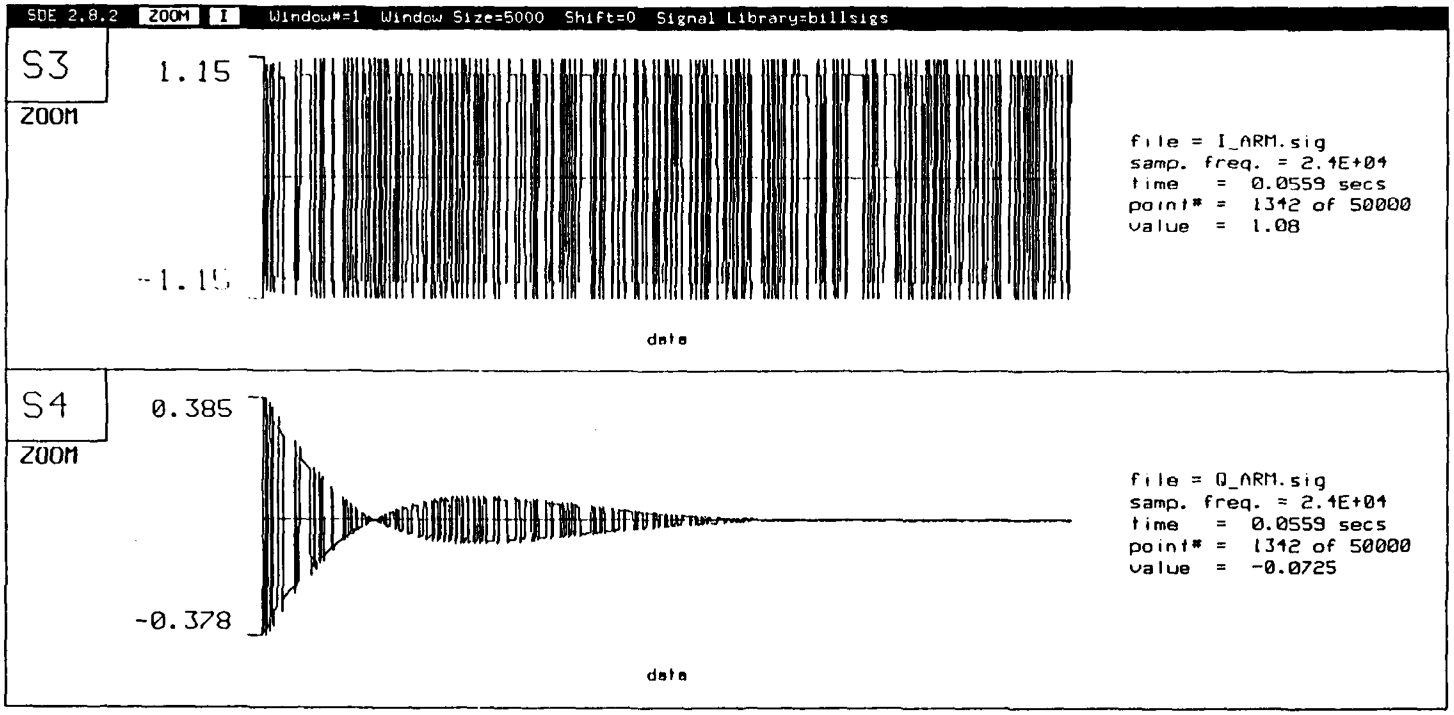

Figure 28. Costas Loop Response to Phase Step Plot 3

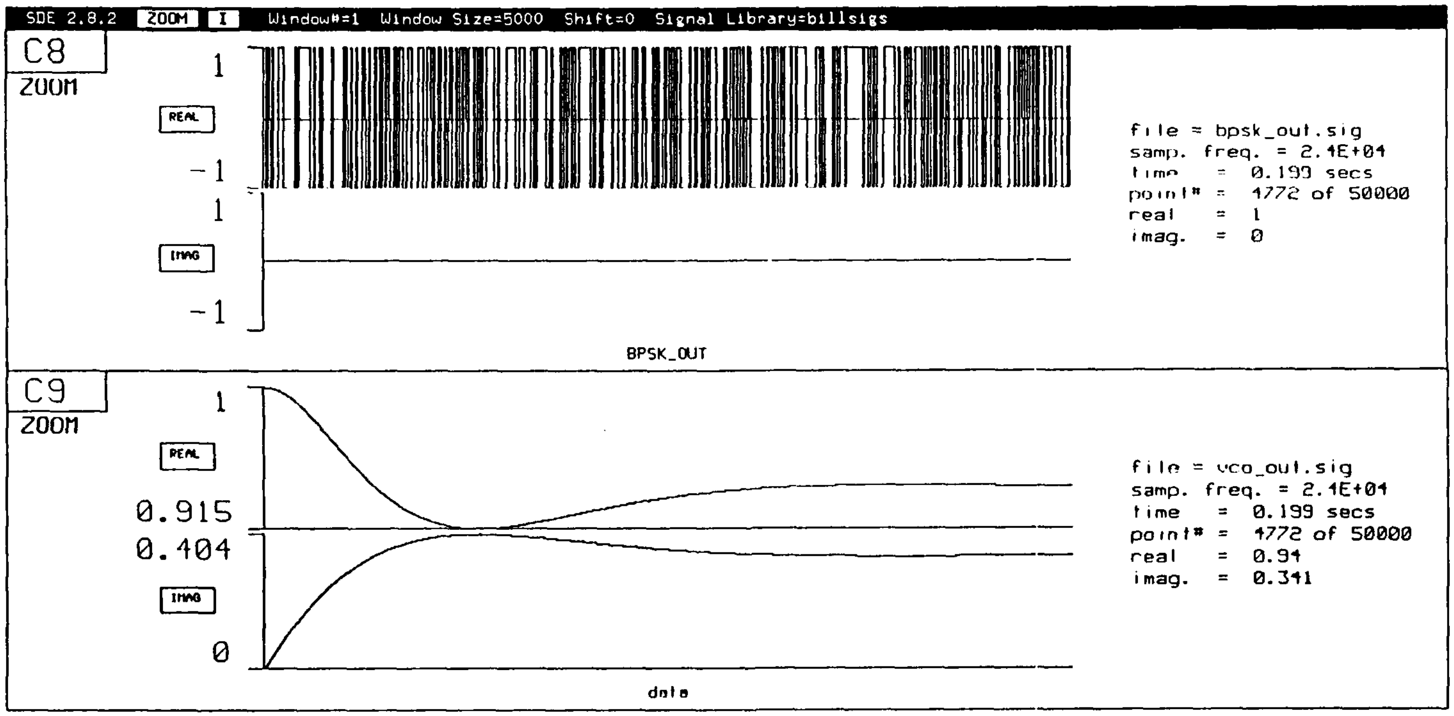

Figure 29. Costas Loop Response to Phase Step Plot 4

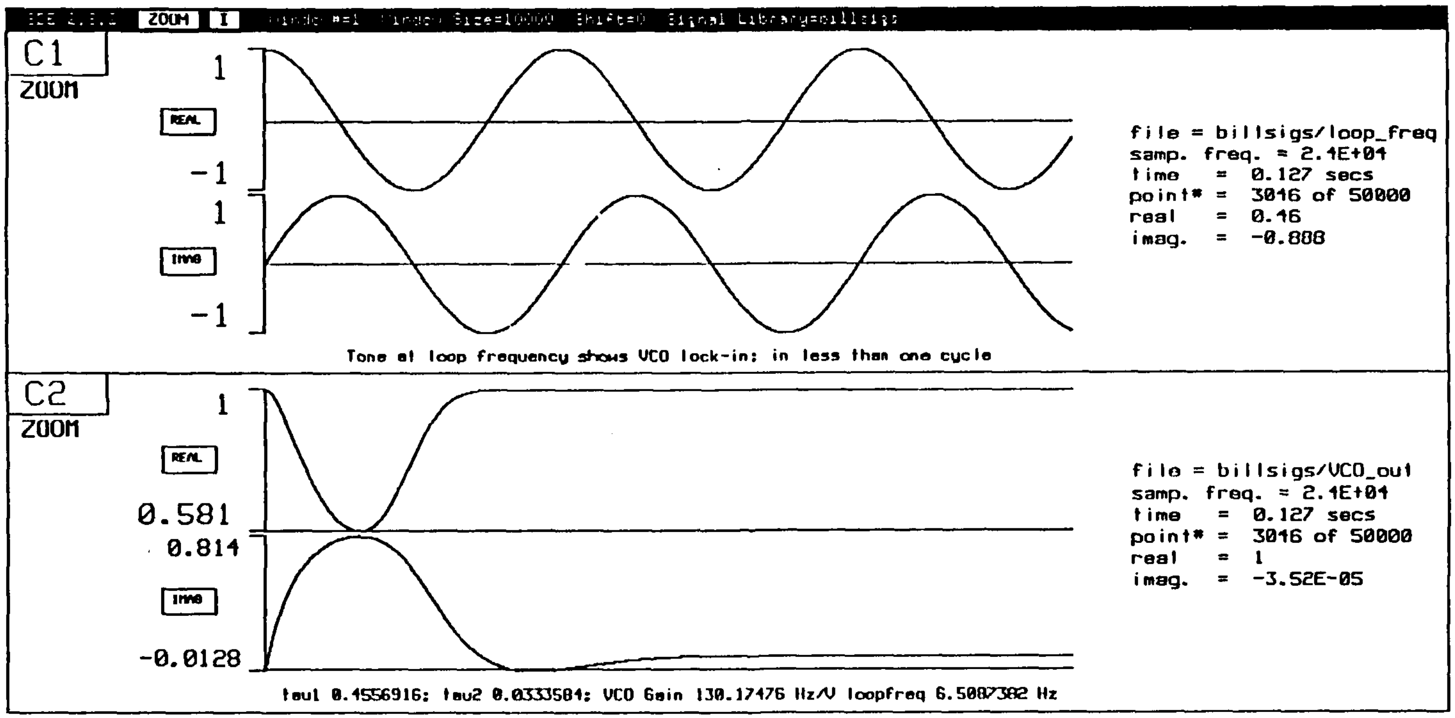

Figure 30. Costas Loop Response to Frequency Step Plot 1

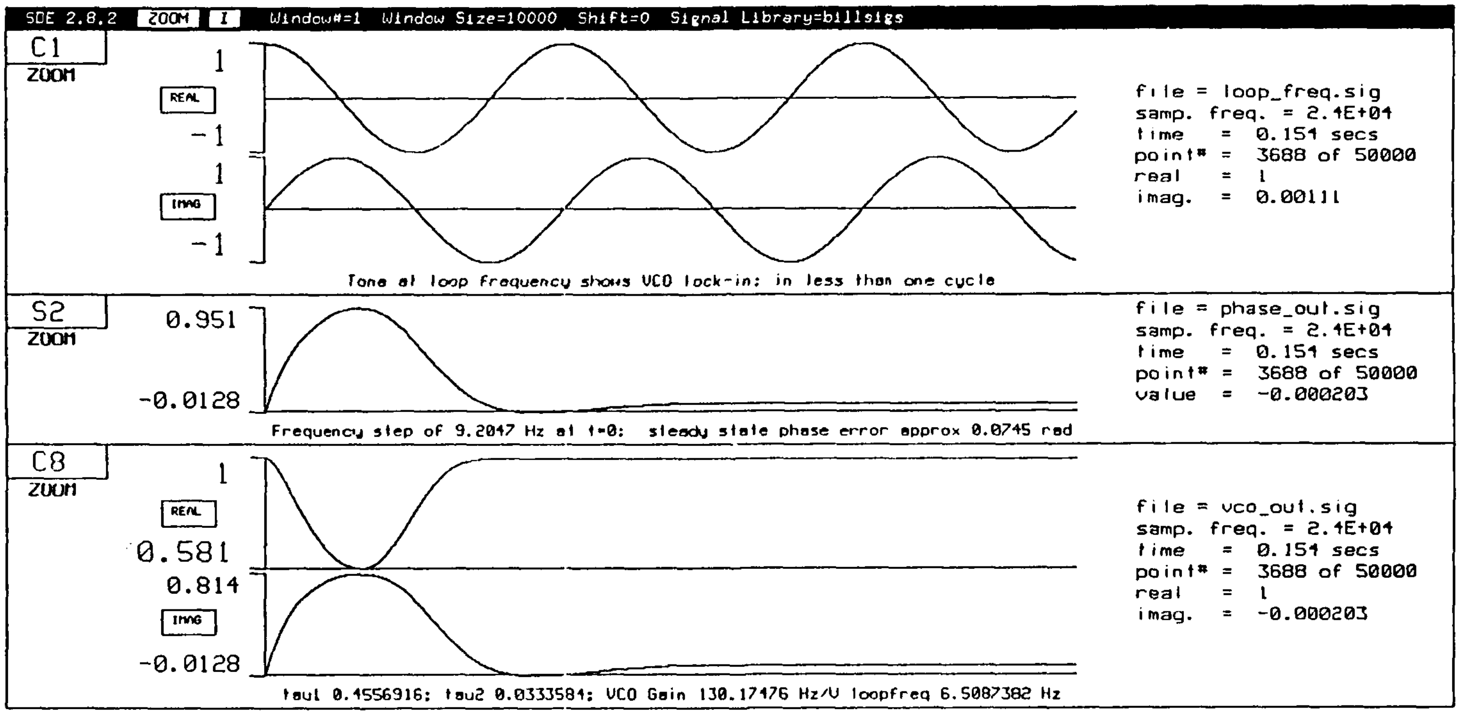

Figure 31. Costas Loop Response to Frequency Step Plot 2

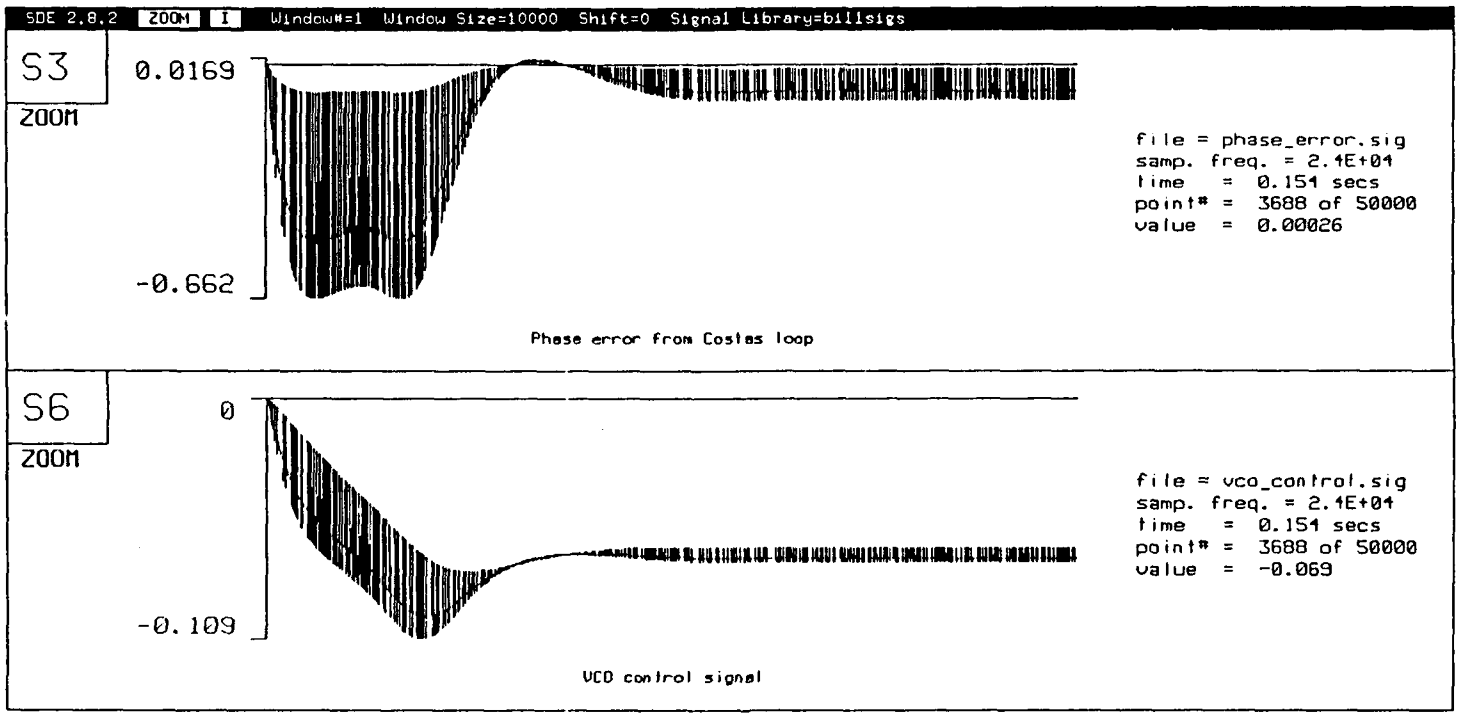

Figure 32. Costas Loop Response to Frequency Step Plot 3

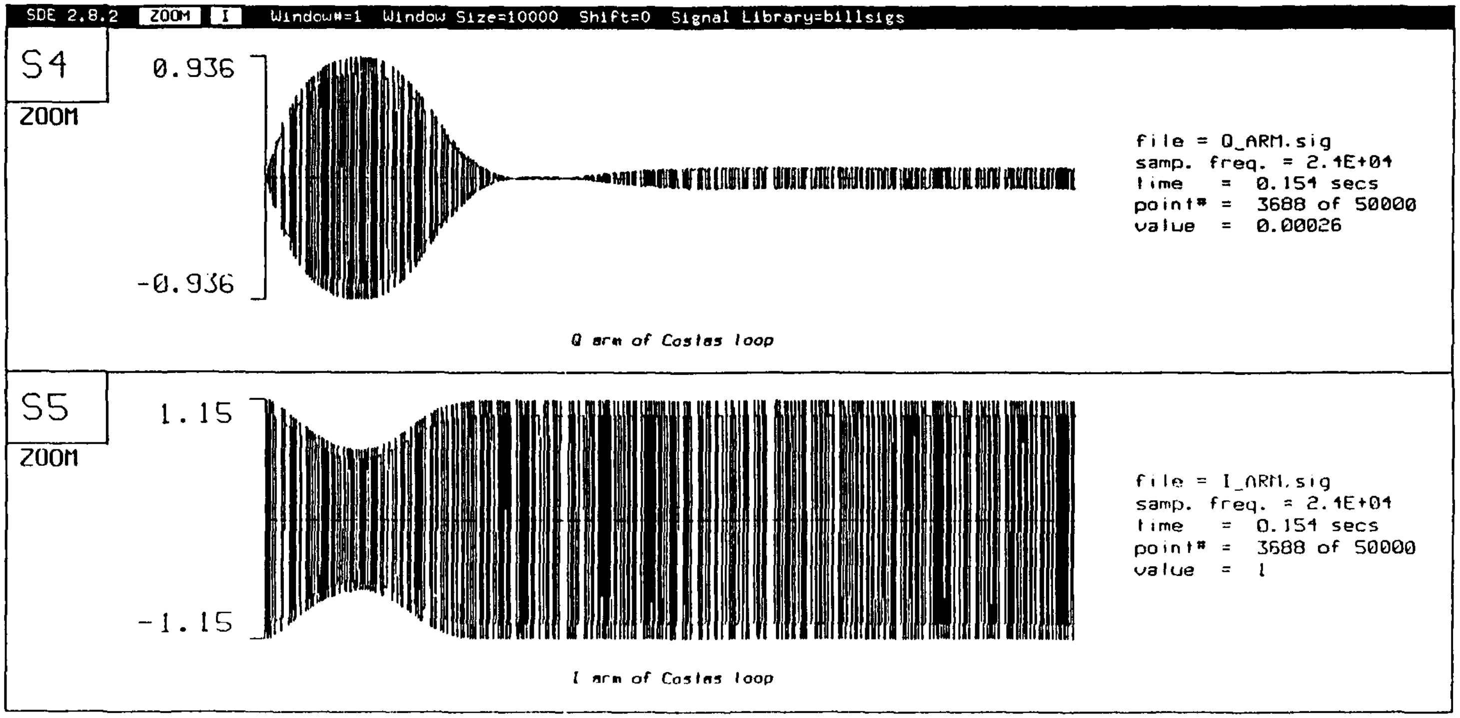

Figure 33. Costas Loop Response to Frequency Step Plot 4

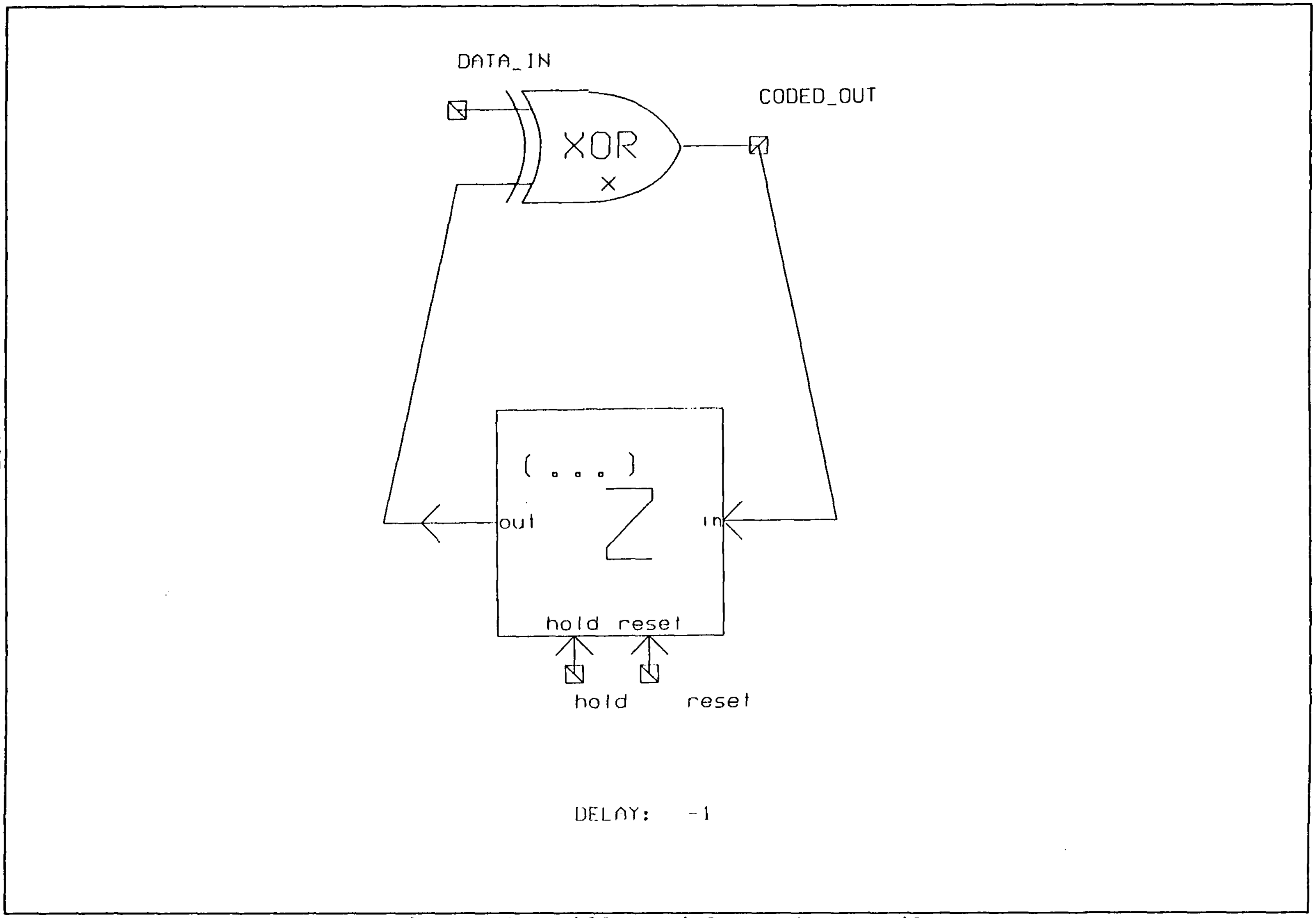

Figure 34. Differential Encoder Detail

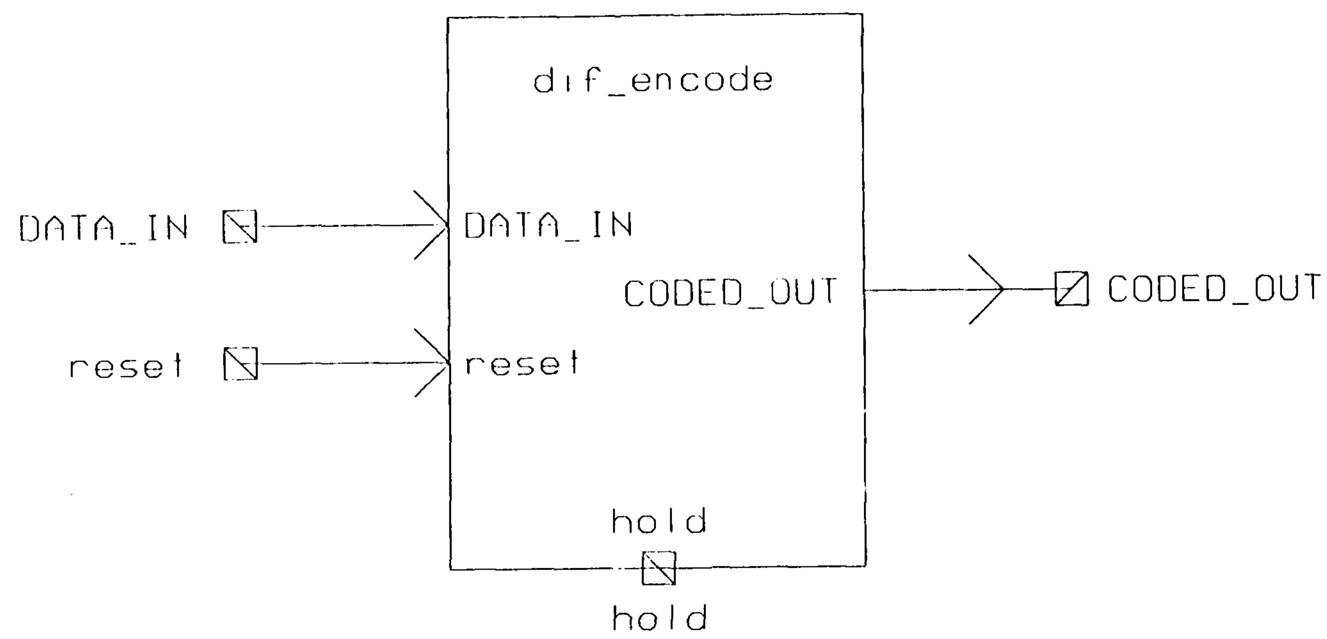

Figure 35. Differential Encoder Symbol

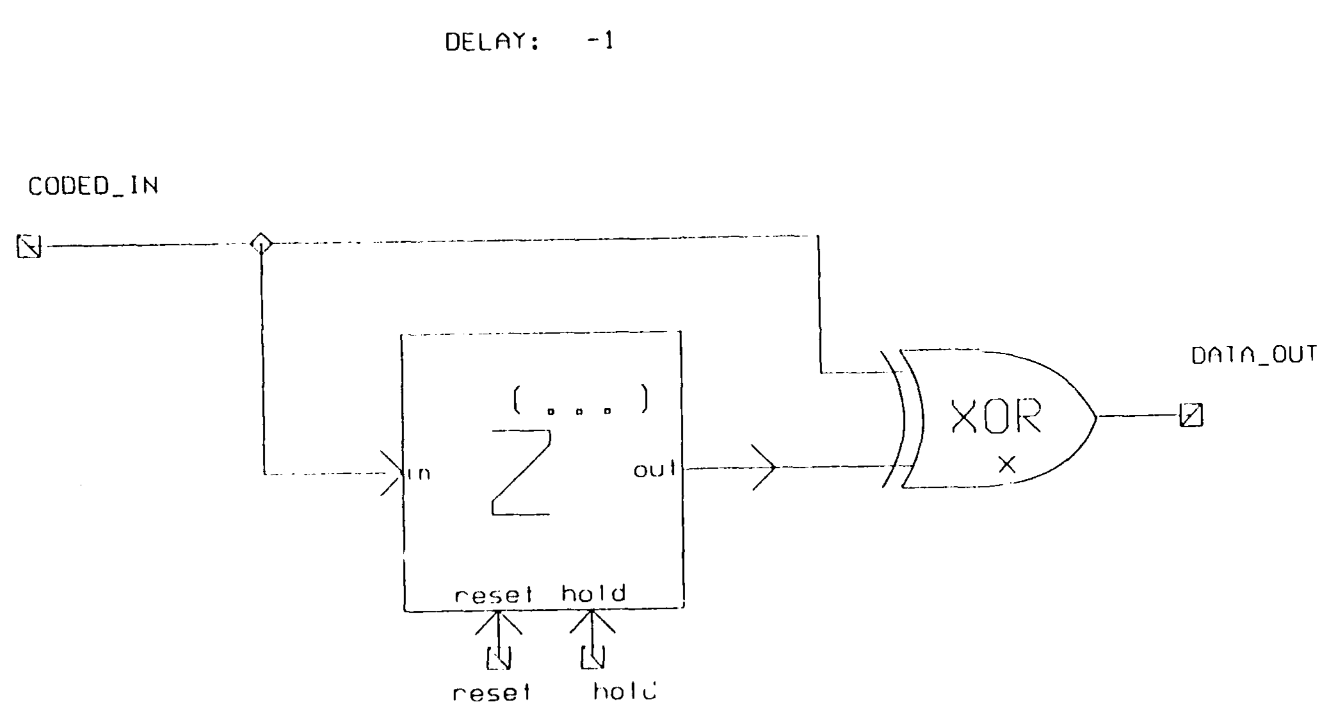

Figure 36. Differential Decoder Detail

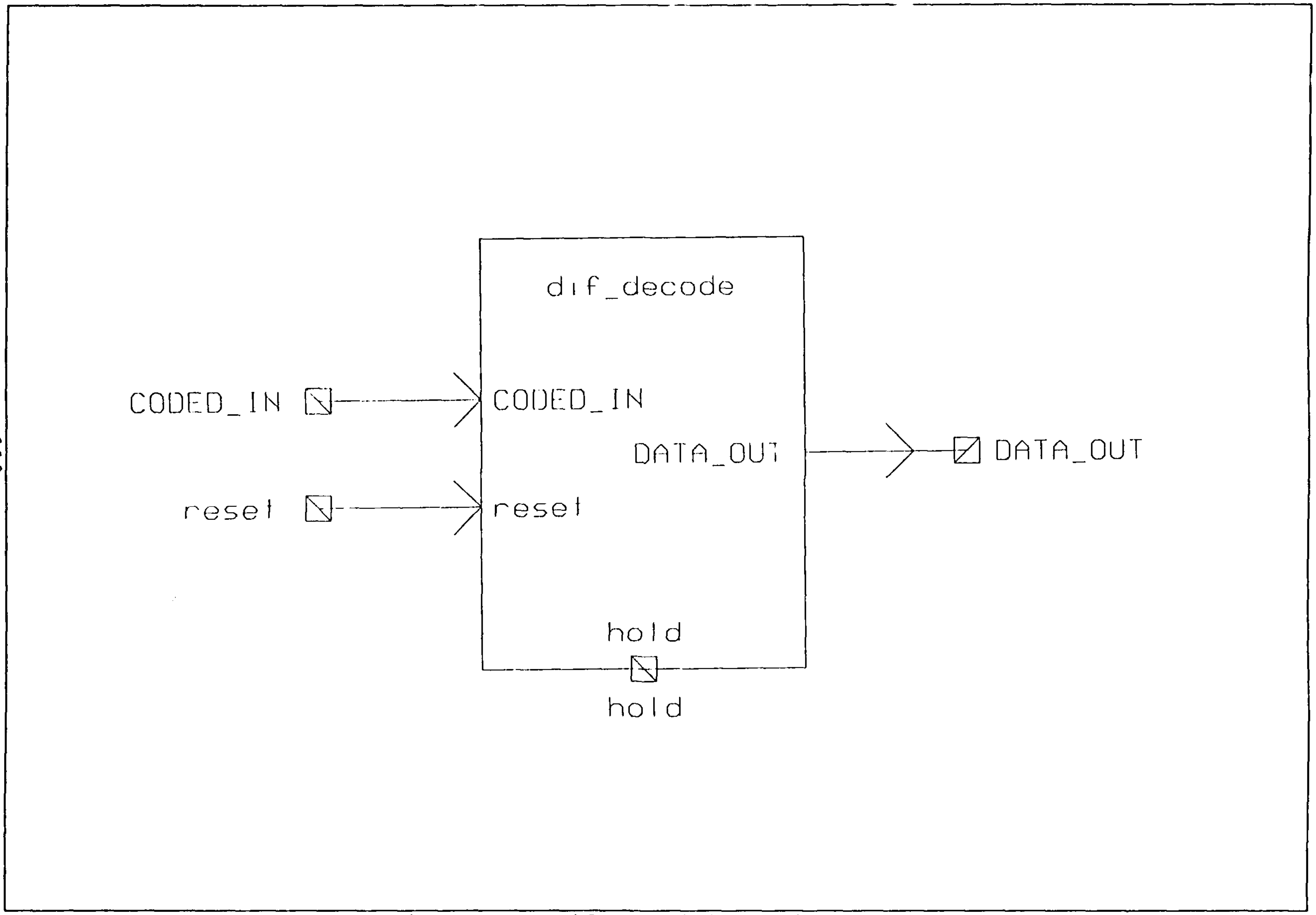

Figure 37. Differential Decoder Symbol

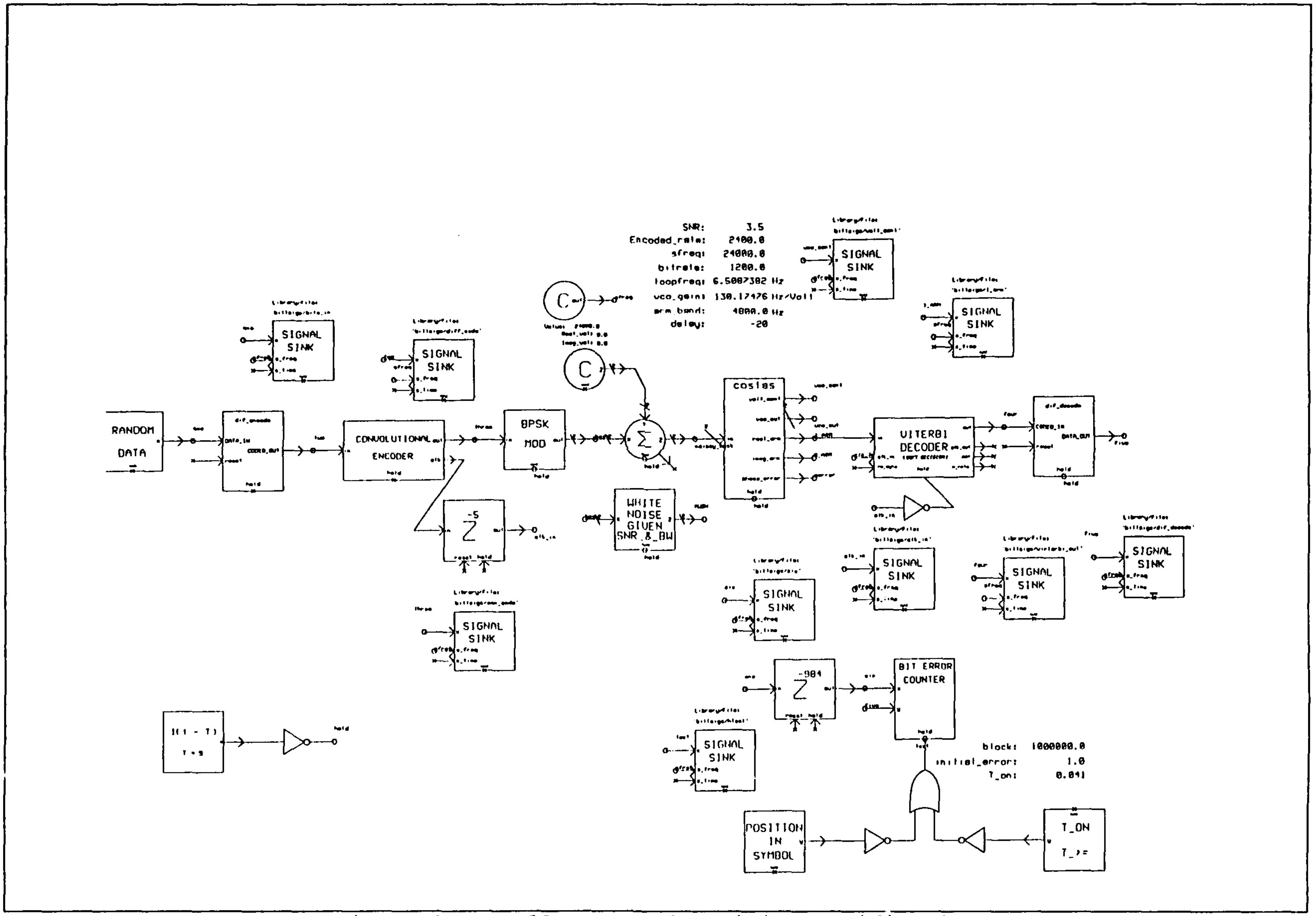

Figure 38. Full System for Timing Verification

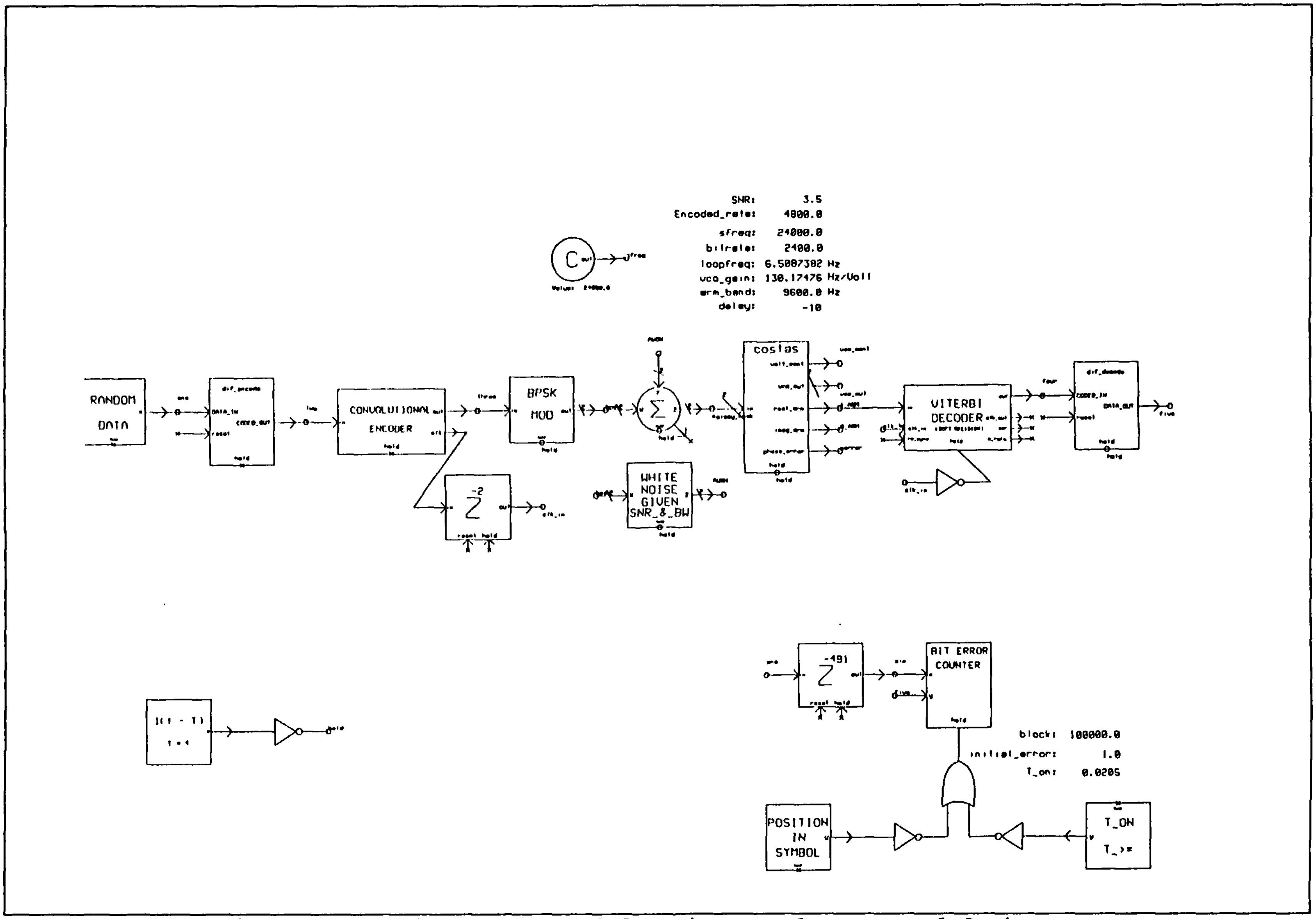

Figure 39. Full System Used for Five Samples per Coded Bit Test

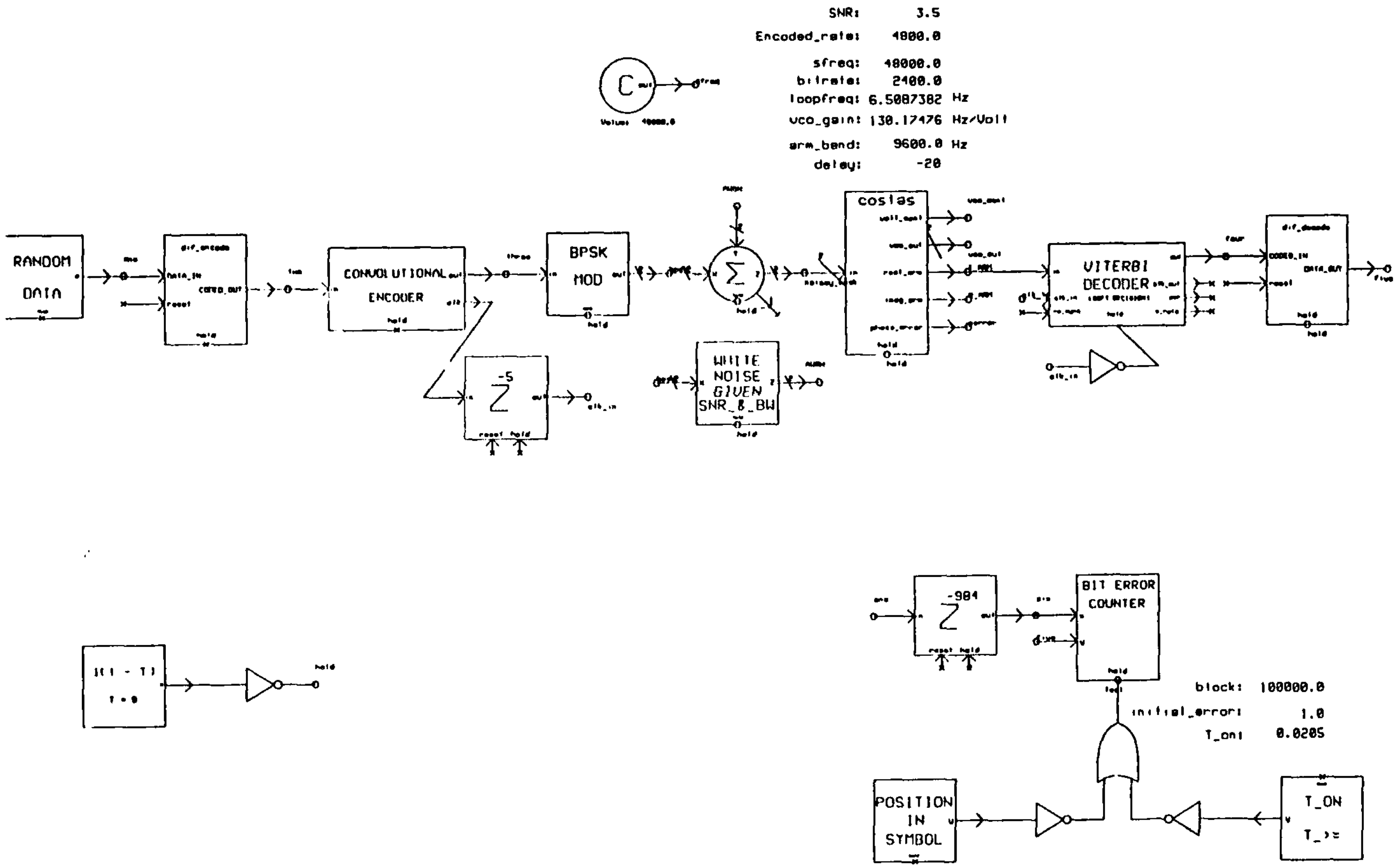

Figure 40. Test Ten Samples per Coded Bit at 2400 Bit per Second

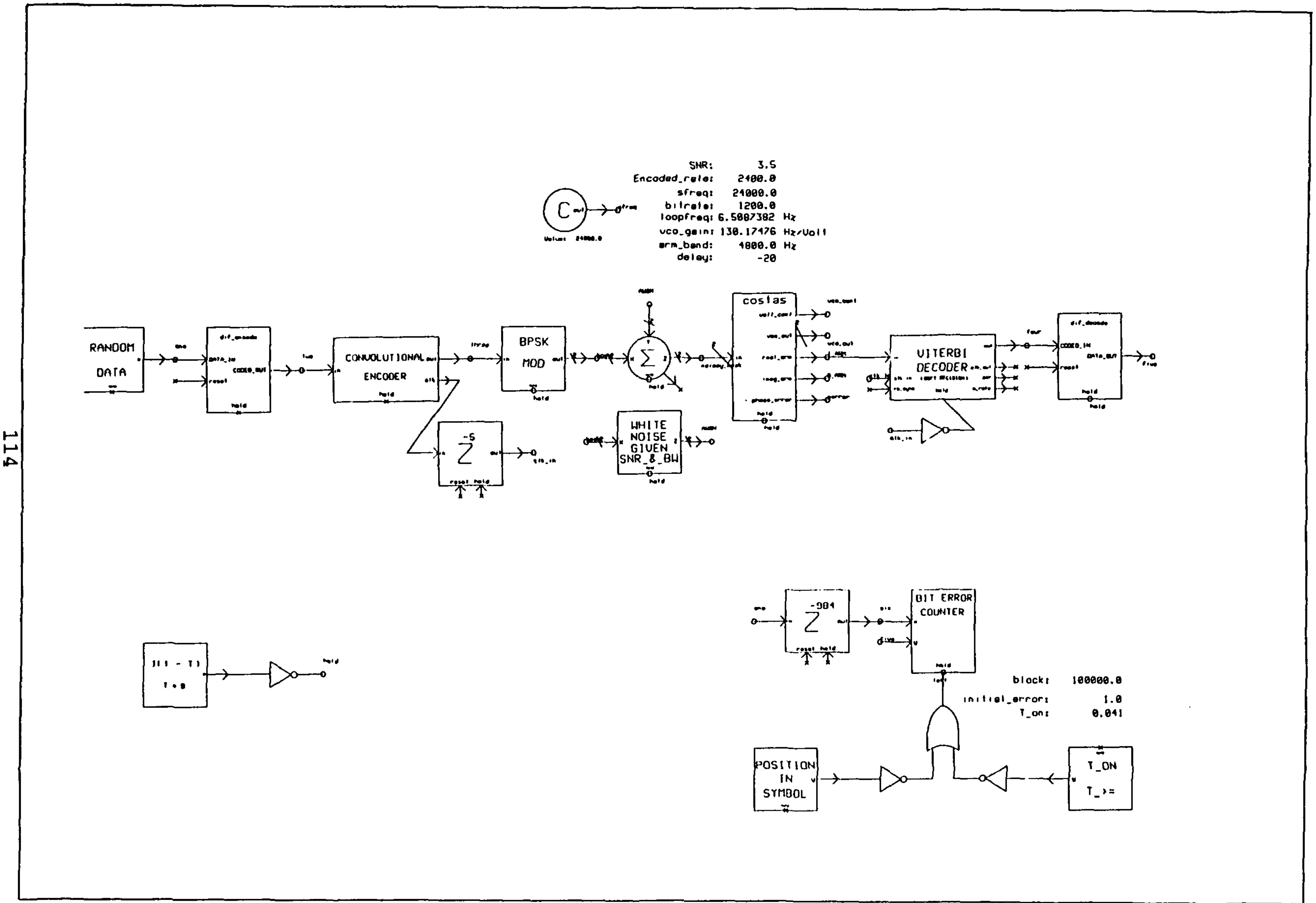

Figure 41. Test Ten Samples per Coded Bit at 1200 Bit per Second

Appendix B

Table 23 through Table 38 are the data results and statistics for estimation of probability of bit error as found using the Monte Carlo Method.

Table 23. $\hat{P}_b$ Matched Filter Detection BPSK For 4 dB E_b/N_o

Run	$\hat{P}_b$ (10^{-2})
1	2.10
2	1.20
3	1.90
4	1.40
5	1.00
6	1.10
7	0.60
8	1.20
9	0.90
10	2.00

Table 24. $\hat{P}_b$ Matched Filter Detection BPSK For 6 dB E_b/N_o

Run	$\hat{P}_b$ (10^{-3})
1	2.40
2	3.60
3	3.20
4	3.50
5	3.20
6	1.70
7	3.20
8	2.50
9	2.90
10	2.20

Table 25. $\hat{P}_b$ Matched Filter Detection BPSK For 8 dB E_b/N_o

Run	$\hat{P}_b$ (10^{-4})
1	2.10
2	2.00
3	2.60
4	2.70
5	2.90
6	3.60
7	1.70
8	2.00
9	2.60
10	2.10

Table 26. $\hat{P}_b$ Statistics Matched Filter Detection BPSK

E_b/N_o	μ	σ^2	σ	ε
4.0 dB	1.34×10^{-2}	2.538×10^{-5}	5.038×10^{-3}	0.403
6.0 dB	2.84×10^{-3}	3.804×10^{-7}	6.168×10^{-4}	0.258
8.0 dB	2.43×10^{-4}	3.157×10^{-9}	5.618×10^{-5}	0.294

Table 27. $\hat{P}_b$ For 5 Samples/Encoded Bit: 3.5 dB E_b/N_o

Run	$\hat{P}_b$ (10^{-3})
1	3.90
2	3.84
3	4.00
4	4.26
5	3.74
6	4.12
7	4.48
8	3.92
9	3.70
10	3.64

Table 28. $\hat{P}_b$ For 5 Samples/Encoded Bit: 4.5 dB E_b/N_o

Run	$\hat{P}_b$ (10^{-4})
1	5.16
2	4.76
3	6.44
4	6.10
5	5.38
6	5.26
7	5.46
8	5.04
9	4.70
10	4.84

Table 29. $\hat{P}_b$ or 5 Samples/Encoded Bit: 5.0 dB E_b/N_o

Run	$\hat{P}_b$ (10^{-4})
1	1.62
2	1.94
3	2.02
4	1.52
5	2.00
6	1.86
7	1.64
8	1.50
9	1.28
10	1.64

Table 30. $\hat{P}_b$ For 5 Samples/Encoded Bit: 5.5 dB E_b/N_o

Run	$\hat{P}_b$ (10^{-5})
1	6.20
2	6.30
3	4.82
4	5.30
5	6.28
6	5.52
7	5.44
8	4.98
9	5.20
10	5.38

Table 31. $\hat{P}_b$ Statistics For 5 Samples/Encoded Bit

E_b/N_o	μ	σ^2	σ
3.5 dB	3.960×10^{-3}	6.95111×10^{-8}	2.63650×10^{-4}
4.5 dB	5.314×10^{-4}	3.24627×10^{-9}	5.69760×10^{-5}
5.0 dB	1.702×10^{-4}	5.97733×10^{-10}	2.44486×10^{-5}
5.5 dB	5.542×10^{-5}	2.89107×10^{-11}	5.37686×10^{-6}

Table 32. $\hat{P}_b$ For 48000 Hz Sample Frequency: 3.5 dB E_b/N_o

Run	$\hat{P}_b$ (10^{-3})
1	1.76
2	2.10
3	1.80
4	2.20
5	2.20
6	1.88
7	2.08
8	1.60
9	1.94
10	1.68

Table 33. $\hat{P}_b$ For 1200 bps Data Rate: 3.5 dB E_b/N_o

Run	$\hat{P}_b$ (10^{-3})
1	1.88
2	1.62
3	1.72
4	1.68
5	2.52
6	1.48
7	1.98
8	1.72
9	1.80
10	2.02

Table 34. $\hat{P}_b$ Statistics For 1200 bps Data Rate : 3.5 dB E_b/N_o

μ (10^{-3})	σ^2 (10^{-8})	σ (10^{-4})
1.842	8.33289	2.88667

Table 35. $\hat{P}_b$ For 10 Samples/Encoded Bit: 4.5 dB E_b/N_o

Run	$\hat{P}_b$ (10^{-4})
1	2.10
2	2.46
3	2.66
4	2.28
5	2.58
6	2.16
7	2.42
8	2.26
9	1.82
10	2.58

Table 36. $\hat{P}_b$ For 10 Samples/Encoded Bit: 5.0 dB E_b/N_o

Run	$\hat{P}_b$ (10^{-5})
1	7.80
2	9.40
3	6.80
4	6.60
5	5.60
6	7.20
7	7.80
8	5.60
9	10.00
10	5.40

Table 37. $\hat{P}_b$ For 10 Samples/Encoded Bit: 5.5 dB E_b/N_o

Run	$\hat{P}_b$ (10^{-5})
1	2.22
2	1.80
3	2.12
4	1.86
5	1.94
6	1.74
7	1.96
8	1.74
9	2.12
10	2.36

Table 38. $\hat{P}_b$ Statistics For 10 Samples/Encoded Bit

E_b/N_o	μ	σ^2	σ
3.5 dB	1.842×10^{-3}	8.33289×10^{-8}	2.88667×10^{-4}
4.5 dB	2.332×10^{-4}	6.75733×10^{-10}	2.59949×10^{-5}
5.0 dB	7.220×10^{-5}	2.47511×10^{-10}	1.57325×10^{-5}
5.5 dB	1.986×10^{-5}	4.49822×10^{-12}	2.12090×10^{-6}

References

1. COM-248 PSK Digital Satellite Modem Installation and Operation Manual. Encinitas, CA: CommQuest Technologies, Inc., 1992.

2. Palmer, Larry C. "Computer Modeling and Simulation of Communications Satellite Channels," IEEE Journal on Selected Areas in Communications, VOL. SAC-2, NO. 1: 89-101 (January 1984).

3. Jeruchim, Michel C. "Techniques for Estimating the Bit Error Rate in the Simulation of Digital Communication Systems," IEEE Journal on Selected Areas in Communications, VOL. SAC-2, NO. 1: 153-170 (January 1984).

4. Jacobs, Irwin M. "Practical Applications of Coding," IEEE Transactions on Information Theory, VOL. IT-20 NO. 3: 305-310, (May 1974).

5. Heller, Jerrold A., Irwin Mark Jacobs. "Viterbi Decoding for Satellite and Space Communications," IEEE Transactions on Communication Technology, VOL. COM-19 NO. 5: 835-848, (October 1971).

6. Trexel, Brian, Engineer. Personal Correspondence and Telephone Conversations. CommQuest Technologies, Inc., June *through September* 1992.

7. Gardner, Floyd M. Phaselock Techniques, New York: John Wiley & Sons, 1979.

8. Sklar, Bernard. Digital Communications, Englewood Cliffs: Prentice Hall, 1988.

9. Yuen, Joseph H. Deep Space Telecommunications Systems Engineering, New York: Plenum Press, 1983.

10. Signal Processing WorkSystem™ User's Guide. Foster City, CA: Comdisco System, Inc., 1990.

11. Jeruchim, Michel C., Balaban, Philip, Shanmugan, Sam K. Simulation of Communication Systems, New York: Plenum Press, 1992.

12. Shanmugam, Sam K., Philip Balaban. "A Modified Monte-Carlo Simulation Technique for the Evaluation of Error Rate in Digital Communication Systems," IEEE Transactions on Communications, VOL. COM-28, NO. 11: 1916-1924 (November 1980).

13. Blum, Julius R., Judah I. Rosenblatt. Probability and Statistics, Philadelphia: W. B. Saunders Company, 1972.

14. Bennett, Capt John B. Implementation of the Modified Monte Carlo Technique Using Importance Sampling on the Block Oriented System Simulator. MS Thesis, AFIT/GE/ENG/90D-03. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1990.

15. Lindsey, William C. Telecommunications Systems Engineering, Englewood Cliffs: Prentice Hall, Inc., 1973.

16. Best, Roland E. Phase-Locked Loops, New York: McGraw Hill, Inc., 1984.

17. Schiavoni, Maryanne T. "Simulation of a QPSK High Data Rate Receiver: Modeling of Tracking Loops and Adaptive Equalizer," IEEE Military Communications Conference-MILCOM90, VOL 2: 679-683 (1990).

Vita

Captain William L. Montgomery was born on 27 December 1960. He graduated from West Forest High School in Tionesta Pennsylvania in 1979. After attending The State University of New York at Alfred, New York for one year, he enlisted in the Air Force in June of 1980. He entered Wright State University, Dayton OH, in August 1983 under the Airman Education and Commissioning Program, and graduated in March 1986 with a Bachelor of Science in Electrical Systems Engineering. He received a reserve commission at Officer Training School in July 1986. After attending the Communication Engineer course at Keesler AFB, he was assigned to Headquarters, Tactical Communication Division, Langley AFB in December 1986. He was assigned as the Facility Engineer for the 1912th Computer Systems Group in January 1988 and oversaw construction of the more than two million dollar addition to the computer facility. He was assigned as Chief of Concepts and Engineering Branch, Headquarters, Tactical Air Command SCOVC, in October 1990. He entered the School of Electrical and Computer Engineering, Air Force Institute of Technology, in May 1991.

Permanent Address: 3109 Bay Landing Dr.
Westlake, OH 44145

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
December 1992

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
Computer Simulation of a General Purpose Satellite Modem

5. FUNDING NUMBERS

6. AUTHOR(S)
Captain William L. Montgomery, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GE/ENG/92D-28

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Mr Jim Foshee
WL/AAAI-1
WPAFB, OH 45433-6543

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The purpose of this research was to model and simulate the performance of a digital phase shift keyed satellite modem. The probability of bit error (P_b) at different levels of energy per bit to noise power ratio (E_b/N_o) was the performance measure. The channel was assumed to contribute only additive white Gaussian noise. A second order Costas loop performs demodulation in the modem and was the key part of the simulation. The Costas loop with second order Butterworth arm filters was tested by finding the response to a phase or frequency step. The Costas loop response was found to be in agreement with theoretical predictions in the absence of noise. Finally, the effect on P_b of a rate 1/2 constraint length 7 convolutional code with eight level soft Viterbi decoding was demonstrated by the simulation. The simulation results were within 0.7 dB of theoretical. All computer simulations were done at baseband to reduce simulation times. The Monte Carlo error counting technique was used to estimate P_b. The effect of increasing the samples per bit in the simulation was demonstrated by the 0.4 dB improvement in P_b caused by doubling the number of samples.

14. SUBJECT TERMS
Modem, Computerized Simulation, Convolution, Encoding, Costas Loop, Coding, Phase Modulation

15. NUMBER OF PAGES
138

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL